NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

SERVICE-ORIENTED ARCHITECTURE APPROACH TO MAGTF LOGISTICS SUPPORT SYSTEMS

by

Frank Sierra

September 2013

Thesis Advisor:	Kishore F. Sengupta
Thesis Co-Advisor:	Alex Bordetsky

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704–0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202–4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704–0188) Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
September 2013

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE
SERVICE-ORIENTED ARCHITECTURE APPROACH TO MAGTF LOGISTICS SUPPORT SYSTEMS

5. FUNDING NUMBERS

6. AUTHOR(S) Frank Sierra

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943–5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ____N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

Legacy logistics systems are an antiquated technology and fall short of providing the Marine Air Ground Task Force (MAGTF) with modern, net-centric, expeditionary Logistics Chain Management (LCM) and Command and Control (C2) capabilities. The Marine Corps owns more than 200 logistics information systems. While some of these systems still perform critical functions, others are stove-piped, redundant, or no longer provide an adequate modern capability. Managing legacy assets and interim technologies while concurrently developing new long-term enterprise solutions is required in order to provide the Marine Corps with the necessary logistics information technology capabilities. The envisioned future end state is logistics data shared across the MAGTF, and ultimately, across the entire organization. A shared-data environment, populated by autonomic computing, will provide actionable logistics data to everyone in the MAGTF, from the "warehouse" to the warfighter position, in near real-time. Common systems supporting common techniques, tactics and procedures which equal significantly improved capabilities. The goal of this research is to envision a set of common information technology capabilities required to execute LCM missions without considering the current limitations provided by existing legacy or MLS2 information technology systems. This research will focus on implementing a service-oriented architecture (SOA) approach to the MLS2 and related processes that will initiate to improve support to the decision-makers and the warfighters across the enterprise. The key end state at hand is to determine a mutually exclusive and comprehensive set of common MLS2 information technology capabilities required to execute C2 for Logistics and LCM's missions.

14. SUBJECT TERMS
Service-oriented architecture (SOA), layers of SOA, interoperability, business process modeling (BPM), business process review (BPR), enterprise resource planning (ERP), enterprise service bus (ESB), logistics chain management (LCM)

15. NUMBER OF PAGES
77

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

THIS PAGE INTENTIONALLY LEFT BLANK

SERVICE-ORIENTED ARCHITECTURE APPROACH TO MAGTF LOGISTICS SUPPORT SYSTEMS

Frank Sierra
Captain, United States Marine Corps
B.A., Northwood University, 2002

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
September 2013

Author: Frank Sierra

Approved by: Dr. Kishore F. Sengupta
Thesis Advisor

Dr. Alex Bordetsky
Thesis Co-Advisor

Dr. Dan Boger
Chair, Information Sciences

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Legacy logistics systems are an antiquated technology and fall short of providing the Marine Air Ground Task Force (MAGTF) with modern, net-centric, expeditionary Logistics Chain Management (LCM) and Command and Control (C2) capabilities. The Marine Corps owns more than 200 logistics information systems. While some of these systems still perform critical functions, others are stove-piped, redundant, or no longer provide an adequate modern capability. Managing legacy assets and interim technologies while concurrently developing new long-term enterprise solutions is required in order to provide the Marine Corps with the necessary logistics information technology capabilities. The envisioned future end state is logistics data shared across the MAGTF, and ultimately, across the entire organization. A shared-data environment, populated by autonomic computing, will provide actionable logistics data to everyone in the MAGTF, from the "warehouse" to the warfighter position, in near real-time. Common systems supporting common techniques, tactics and procedures which equal significantly improved capabilities. The goal of this research is to envision a set of common information technology capabilities required to execute LCM missions without considering the current limitations provided by existing legacy or MLS2 information technology systems. This research will focus on implementing a service-oriented architecture (SOA) approach to the MLS2 and related processes that will initiate to improve support to the decision-makers and the warfighters across the enterprise. The key end state at hand is to determine a mutually exclusive and comprehensive set of common MLS2 information technology capabilities required to execute C2 for Logistics and LCM's missions.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

ATLASS	Asset Tracking Logistics and Supply System
BCS3	Battle Command Sustainment Support System
BPM	Business Process Modeling
CLC2S	Combat Logistics, Capability Support System
COP	Common Operating Picture
COTS	Commercial of the Shelf
C2	Command and Control
DC I&L	Deputy Chief of Staff Marine Corps Installations & Logistics
DoD	Department of Defense
ERP	Enterprise Resource Planning
ESB	Enterprise Service Bus
GCSS-MC	Global Combat Support System-Marine Corps
IT	Information Technology
KPI	Key Performance Indicators
LCE	Logistics Command Element
ITV	In-transit Visibility
LCM	Logistics Chain Management
MAGTF	Marine Air Ground Task Force
MLS2	MAGTF Logistics Support Systems
NFR	Nonfunctional Requirements
SASSY	Supported Activities Supply System
SLA	Service Level Agreement
SOA	Service-oriented Architecture
TAV	Total Asset Visibility
TCPT	Transportation Capability Planning Tool
TECOM	Training and Education Command
TLOC	Tactical Logistics Operations Centers
WSDL	Web Services Description Language

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

First and foremost, I desire to thank God for the many blessings he has given me. The opportunity to advance my education at Naval Postgraduate School has truly been a great blessing. Very special thanks to my wife, Victoria Sierra, who has given me her unconditional love and support throughout my tenure at NPS as well as the past 17 years of my Marine Corps career. Her unselfish sacrifice and endless encouragement allowed me to successfully complete my studies and thesis. To Audrie, Xannria and Frankie, I am privileged to have been blessed with such great children. My family's unconditional love and support has motivated me to pursue greater accomplishments, and it's because of them that I drive toward higher ground. My steadfast integrity and commitment is dedicated to the Marine Corps. The Marine Corps has presented me with many opportunities, and I am grateful that NPS has been one of them. My extended family is the Marine Corps, and I feel proud to be part of such an awesome organization. I want to thank my thesis advisors, Dr. Kishore F. Sengupta and Dr. Alex Bordetsky, for their time, guidance and inspiration in supporting me throughout this entire thesis process. Dr. Kishore F. Sengupta, thank you for your genuine interest in my thesis and for all the extra sessions that taught and mentored me in my research. Dr. Alex Bordetsky, your subject matter expertise was a remarkably insightful and you were keen with sanity checking my thesis work. NPS afforded me an incredible opportunity to earn a master's in information technology management. I owe a tremendous amount of my success to all my outstanding professors, the Marines and sailors in my cohort, and my friends that supported me along my educational path.

"*Education is the most powerful weapon which you can use to change the world.*"

—Nelson Mandela

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

As a result of technological developments, the Marine Corps has sought out new and improved approaches to conducting Logistics Chain Management (LCM) processes. Systems have been designed to augment and manage core business functions such as supply, maintenance, accounting, procurement, and distribution. However, even with these systems in place, information is unreliable and inconsistent if they are on disparate platforms. It is not uncommon for organizations throughout the Marine Corps to have implemented a wide range of distinct technologies. Functioning with a wide range of disparate systems that do not interface or are integrated forces users to spend valuable time performing laborious data manipulation tasks. Furthermore, these disparate platforms challenge users with accessing, sharing, understanding and awareness of useful data, as well as identifying pressing demands for actionable information which often results in approximations offered to decision makers.

The reality is that organizations cannot afford to throw away all of their existing legacy applications; rather they must leverage their existing investments. The challenge is obtaining an overall perspective for the hundreds of disparate systems which provide a complex global-scaled logistics capability to the Marine Corps. Coordinating and integrating the right data at the right time and place on such a global scale is very complex. Additionally, our conflicts in Iraq and Afghanistan have fielded many stand-alone software systems without much thought as to how effectively they would share information within networks. The plethora of logistics information systems has overwhelmed tactical logisticians and in most cases the systems were redundant, complex and specific to only one functional area.

To ensure interoperability throughout the Marine Corps Logistics Chain Management, the architecture should be redesigned from a holistic view. The current systems were designed primarily from the functional user’s perspective which is why many of the automated information systems are not interoperable. An extra effort should be made to ensure that data are not disjointed or systems designed solely from the narrow perspective of an individual agency or functional area such as supply, transportation, or finance.

B. RESEARCH OBJECTIVES

The purpose of this research is to determine an alternative MAGTF Logistics Support Systems (MLS2) information technology architecture service necessary to execute C2 for Logistics and LCM’s missions. The benefits of this research complements Deputy Chief of Staff Marine Corps Installations & Logistics (DC I&L) enterprise-level goals which define the infrastructure required to integrate services and business entities across the MAGTF. This study will contribute in defining a Service Oriented Architecture (SOA) approach between MLS2s in order for logistic organizations to better execute its missions.

C. RESEARCH QUESTIONS

This thesis provides the decision maker with the following answers as well as recommendations for future studies.

1. Research Question: How can a Service Oriented Architecture (SOA) approach lead to improved MLS2 IT architecture coordination required to accomplish C2 for Logistics and LCM’s missions?

2. SOA approach to MLS2 allows for an evaluation of current and relevant technologies which align with mission and business goals rather than the availability of future capabilities. A SOA implementation would support the critical requirements for LCM’s critical mission requirements and would ensure performance, availability and interoperability. Integration with both legacy and new technologies is recognized within a SOA approach making this architecture a flexible implementation method.

D. METHODOLOGY

This thesis will focus on industry and DoD reports which provide analysis and benefits of implementing a SOA approach. Business Process Review (BPR) and system design methods will be used to analyze results; including but not limited to the following techniques; modeling and analysis of data; methods for measurement, experimental control and manipulation of variables; collection of empirical data; and interviews.

E. SCOPE

The scope of this thesis is to primarily identify and give recommendations on information system capabilities to execute MLS2 missions. The report will contain a description of all the work conducted, all the models, and the analysis.

Availability and suitability of source data from MLS2s—any approach to understanding the capability needs for Global Combat Support System -Marine Corps (GCSS-MC) will need to perform some form of data collection. It’s assumed that GCSS-MC will have documentation in the form of policies and procedures, existing documentation in the enterprise IT infrastructure, strategic plans, annual reports, and other documents that will provide information on the processes, organization structure, and information needs of MLS2s.

F. THESIS ORGANIZATION

This thesis is organized as follows:

- Chapter I is the introduction of the thesis.
- Chapter II provides background information on SOA standards.
- Chapter III describes the current process design to a set of MLS2.
- Chapter IV presents an alternative architectural design and set of new services to improve its degree of service-orientation.
- Chapter V summarizes the thesis and makes recommendations for future research.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND

A. INTEROPERABILITY PROBLEMS

Interoperability in systems ensures proper communication in heterogeneous environments to increase service usability (Marks, 2006). Currently, the tactical logistics operations center's (TLOC) software systems for expeditionary Logistics Chain Management (LCM) and Command and Control (C2) across the Marine Corps lack interoperability. LCM systems have been built without entirely understanding of how they will connect with other systems. Therefore, this particular system requirement has resulted in the following critical shortfalls throughout the overall IT architecture; (1) poor standardization across the MAGTFs; (2) stove-piped, overlapped and duplicated systems' functionality; and (3) an unknown or unforeseen prerequisite to interface newly introduced technology applications with existing legacy systems.

Standardization is essential in order to attain LCM's most critical mission which is to provide global, integrated logistics management capability in support of the operating forces to maximize their readiness and sustainability. Additionally, the lack of interoperability for TLOC operating systems is a challenge for logistics command element (LCE) customers and LCE combat logistic regiments and battalions shops of these war fighting units. At the operating level, we currently find a menagerie of TLOC operating systems. Combat Logistics, Capability Support System (CLC2S), Transportation Capability Planning Tool (TCPT), Battle Command Sustainment Support System (BCS3) and legacy systems such as Supported Activities Supply System **(**SASSY) are a few examples of the leading operating systems and later addressed in Chapter III. In execution of these systems, LCEs utilize some, none or all of these technologies, thus creating significant inefficiencies due to the lack of unity of effort specifically when building MAGTFs sourced from multiple MARFORs. This issue negatively impacts ground, air combat and air element units that request lateral logistic support beyond their organic capability. The lack of standardization carries over the inability to effectively have unity of effort which makes it unmanageable to provide functional battalions, CLBs or Detachments with TLOC capabilities that merge efficiently into existing TLOC IT systems.

Along with the lack of standardization across the IT architecture of tactical logistics support, MLS2 technologies demonstrate characteristics of stove-piped systems. Most of the systems dealt with today have gone along the path of building their own complete infrastructure and their own hardware and software protocols. Current systems show a large gap between architecture documentation and implemented software which causes unfamiliar key aspects of integration solutions. The lack of architectural vision causes users to invent workarounds which obligates MAGTF logistic users and customers to ingeniously create interfaces between multiple disparate logistic systems. Ownership and management of such heterogeneous systems is increasingly difficult when system modifications are introduced. The key problem, with regard to interoperability with stove-piped systems is the lack of common multisystem conventions. Stove-piped systems are integrated in an ad hoc manner using multiple integration strategies and mechanisms. For example, subsystems are integrated point to point, thus the integration approach for each pair of subsystem is not easily leveraged toward that of other systems. Furthermore, the system implementation is fragile because there are many implicit dependencies upon system configuration, installation details, and system state. The system is difficult to extend, and extensions add additional point-to-point integration links. As each new capability and alteration is integrated, system complexity increases throughout the life cycle of the stovepipe systems; subsequently, system extension, configurability and maintenance become increasingly inflexible (Brown, 1998).

Another issue to recognize is the fact that the Marine Corps has implemented new technology initiatives that require interfaces with current legacy IT platforms. Many of the systems that currently operate the TLOC's IT environment date back to the 1960s and remain the core of the IT portfolio. Legacy systems have survived mergers, acquisitions, re-engineering efforts, technical revolutions, industry realignment and so on. These legacy systems tend to limit TLOC's information sharing capabilities. Legacy systems are considered to be potentially problematic because they are obsolete and increasingly difficult to maintain, improve, and expand. Integration with newer systems may also be difficult because new software may use completely different protocols and technologies. The circumstance of dealing with the integration of both antiquated and new systems is something that must be currently dealt with as the DC I&L attempts to develop long term enterprise solutions.

As technology becomes more widespread and systems become interconnected, interoperability has become essential. Joint Vision 2020 states; "Interoperability is the foundation of effective joint, multinational, and interagency operations" (Joint, 00, p 15). The Marine Corps has developed and implemented numerous independent and redundant MLS2 IT systems which have created fragmentation within the organization's IT architecture. Commands are being forced to maintain an extensive IT portfolio, which requires comprehensive management in order to develop a common situational picture and to accomplish C2 for logistics and LCM's missions. These systems fall short of providing the MAGTF with truly modern, net-centric, expeditionary logistics capabilities.

This chapter introduces SOA concepts and definitions so readers new to the subject can put the material presented in the remaining chapters into the proper context.

B. SERVICE-ORIENTED ARCHITECTURE

Service-oriented Architecture (SOA) is an architectural style that supports service-orientation. Service-orientation is a way of thinking in terms of services and service-based development and the outcomes of services. SOA allows business and information technology merging through an agreement on a set of business-aligned services that collectively support an organization's business processes and goals. For the Marine Corps, it is particularly important to share information in order to provide timely and accurate data for decision makers. The ability to couple components in multiple configurations within the structure of a framework is the primary benefit of SOA. Interoperability and coherence is achieved when you get a system that does what you want it to do (Hayes-Roth, 2003).

Data interoperability is supported by making data assets understandable and by enabling business and mission processes to be reused where possible. SOA is an enabler for interoperability. A SOA solution can provide the LCE entities with shared information to gain situational awareness in order to attain information superiority and therefore achieve outstanding support of the operating forces. As such, organizational leaders should evaluate an IT strategy based on its ability to facilitate SOA. SOA is based on the optimization of information sharing and exchange to facilitate interoperability and performance at the enterprise level rather than the entity level (Marks, 2006).

1. Principles of SOA

The basic principle of SOA is applicable in the entire enterprise architecture. The principle of service orientation is generally applied to the organization of software that maintains the enterprise’s business operations. SOA organizes such software to a set of software services. These services are maintained by an infrastructure together with the services which make improvements on information flow within the business enterprise and other external enterprises. SOA solution stack allows business enterprises to reuse the current applications and technologies while aggregating interoperability, flexibility and agility (Erl, 2007). Flexibility and agility are facilitated because automated business processes and their service elements can be modified without re-coding applications or deploying a new infrastructure to support these rapid technological changes. Interoperability will transform a current manually intensive business process into an automated, adaptive and quick method. In SOA, data and business logic are automated in modular business components with documented interfaces. This clarifies design and facilitates gradual development, it also allows for future extensions. The common set of principles that allow SOA applications to become the solution to integrate diverse, external legacy and commercial of the shelf purchased applications are included in Table 1.

Table 1. SOA Principles

Principle	Rationale
Standardized	
It is the description language that defines service interactions SOAP (Simple Object Access Protocol) web services are gaining ground as the most used implementation of SOA.	➢ SOA manages two computing bodies for instance programs that interact in a manner that enables one entity to execute a unit of work on behalf of another entity ➢ With the protocol independence of SOA, consumers are free to communicate with the service in different methods ➢ It is advantageous when there exists a management layer between the consumers and the service providers in a move that will complete flexibility concerning execution protocols where services conform to a service description. ➢ The aspect of standardization ensures that there are quality management processes and services while improving existing and new properties in a network. ➢
Loose Coupling	
Loose coupling is the idea normally used to deal with requirements of fault tolerance, scalability and flexibility	➢ The objective of loose coupling is to reduce dependencies. The lower the dependencies, the fewer the consequences of modifications to or faults in another system
Service Abstraction	
Services conceal the logic they encrypt from the outside world. By acting in such manner, services enable and preserve the initially described loosely coupled bond	➢ Service abstraction institutes important role in the design and positioning of service compositions.
Service Reusability	
In order to maximize reuse, logic must be divided into services. The value of service Reusability lay emphasis on service positioning as enterprise resources with dubious functional context	➢ The value of service Reusability lay emphasis on service positioning as enterprise resources with dubious functional context.
Service Autonomy	
This principle raises various concerns that relate to the design of service logic and the service's real execution environment. Service normalizations and isolation levels considerations are taken into justification to achieve appropriate measure of independence, particularly for reusable services that are frequently pooled	➢ The value of service independence supports the degree to which other design values can be effectively articulated in real world production spheres by nurturing design features that increase a service's behavioral predictability and reliability.
Service Statelessness	
In any case, services should be stateless. Services are intended to remain stateful on demand. For principle of Service Statelessness to be applied on realistic grounds, statelessness must be assessed first based on availability of adjacent technology architecture that provides state management delegation and rescheduling options	➢ Excessive state information if not managed can compromise service adequacy and weaken scalability potential

Principle	Rationale
Service Discoverability	
Services are discovered in the service registry	➢ Services that are positioned as IT assets with repeatable return on investment need to be easily recognized and understood when chances for reuse present themselves
Service Composability	
The complexity of fundamental service composition alignments increase in complexity as the sophistication of service oriented solutions continue to grow.	➢ The principle of Service Composability deals with this necessity by guaranteeing a variety of concerns taken into account. ➢ The capacity to ultimately compose services is a vital requirement for realizing some of the most paramount objectives of service oriented computing. ➢ Sophisticated service compositions task on service design need to be foreseen to avoid mass retro-fitting efforts.
Service Interoperability	
All the principles of SOA contribute to service interoperability in a way	➢ Interoperability is applied to ensure standard approaches to communication ➢ The identified services can be used by wider audiences hence making the business abilities reusable void of any impact on specific platform application interfaces

2. Layers of SOA

The architectural figure shown in (Figure 1) represents SOA as an array of logical layers. The design of SOA has a nine-layer solution stack which reinforces SOA business value. Each layer of SOA has two attributes: logical and physical. The first attribute which is logical aspect is composed of the entire architectural building blocks, options, KPI (key performance indicators), design decisions and the corresponding; the physical attributes which is the second attribute of single layer covers the comprehension of single logical aspect utilizing products and technology (Aziz, 2006). The solution stack functions to tally the fundamental elements of individual SOA solution. It additionally provides architectural base for the solution.

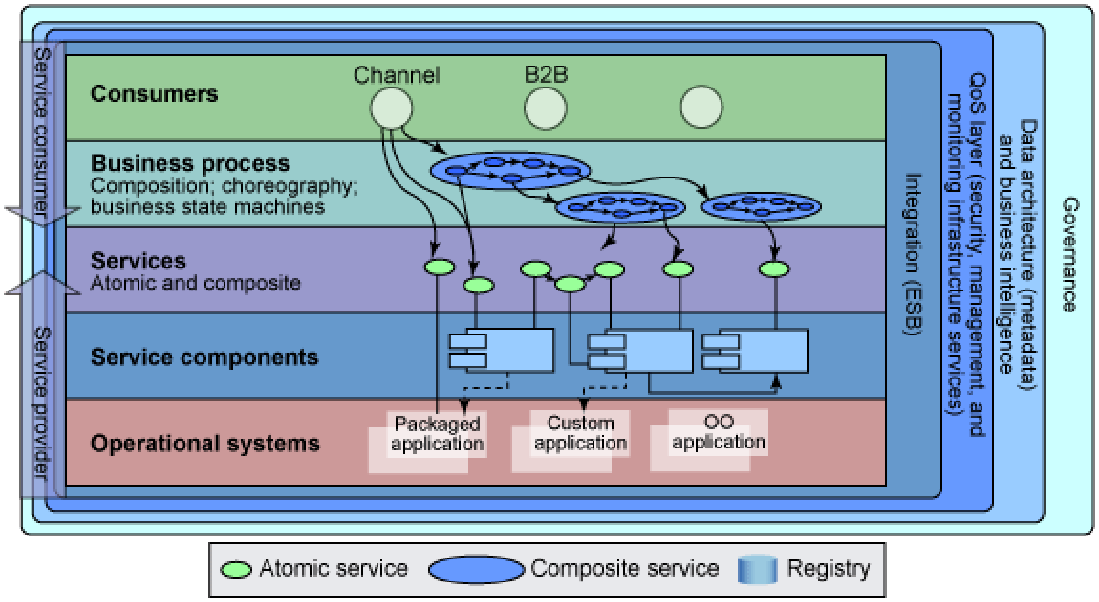


Figure 1. SOA Solution Stack Model (After Thomas, 2004)

a. Layer 1 (Operational Systems)

Layer one consists of all personalized or packaged application properties in the application range. It runs in IT operating system where it supports business activities. This layer delineates the deployment infrastructure and the runtime. These two properties are composed of programs, application servers, platforms, runtime environments, containers, and packaged applications, virtual machines, among others that are installed on the hardware and are required to support the SOA solution. Since the operation layer is comprised of present software application systems, it functions to influence the current IT investments in executing SOA solution (Thomas, 2004). This layer determines directly the overall expenditure of executing the SOA solution. This is important because the layer assists in freeing up the budget for new developments and initiatives for the established business-critical services.

b. Layer 2 (Service Components)

Layer 2 is primarily software components. Each software component provides execution of, realization of, or procedure on a service. The definition of service is reflected by service components, both in the quality of service component and its functionality. The service component layer is aligned to service contracts which are specified in the service layer; it assures the conformity of IT execution with service outline. In terms of “faithful” service realization, the service component layer is considered enforcement. This guarantees quality of service as well as devotion to service-level agreements (SLAs) (Erl, 2007). The service component layer is a master of business flexibility. Through this function, it supports execution of IT malleable services with their layering and composition.

c. Layer 3 (Services)

All the services specified between SOA are integrated in layer 3. This layer is horizontal in alignment and provides the business functions as supported in the SOA. When SOA is introduced, the service layers instigate the notion of services which are purposefully outlined interfaces for capability into the architecture. For the function of this position architecture, a specific service is deliberated to be a theoretical condition of a collection of (either singular or more) business related IT functions. The condition informs consumers with adequate details to petition the business roles exposed by service provider; logically this is performed in an autonomous platform. The service conditions are inclusive of policy documents, attachments that group or indicate service dependencies and SOA management explanations (Erl, 2007). In service layers, there are noteworthy successor-predecessor relationships that exist between layers. This is to say that some of the notable services in the layer 3 may be forms of other services.

Exposed services exists in service layer; they can be identified and raised or be customized to establish a complex service (Graham, 2004). Services are utilities that are available across a network via distinct crossing points of the service layer. The service layer also incorporates enterprise-scale components, project specification components, business-unit components and externalizes a subdivision of their interfaces in a manner of service descriptions. In a nutshell, the components deliver services via their interfaces. Interfaces are conveyed as service descriptions in service layer: services exist as composite or isolation.

Figure 2 shows a magnified service layer; it also depicts how the service layer can be divided into subsets. It is comprised of services that are supplied by a given architecture, which includes both the atomic and composite services.

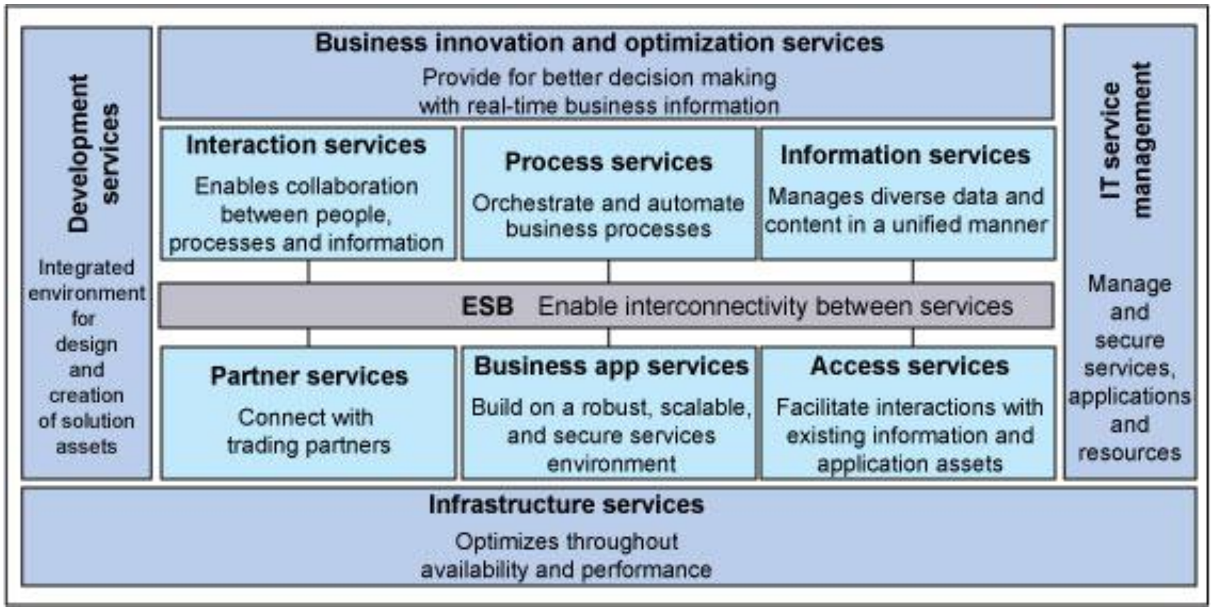

Development services
Integrated environment for design and creation of solution assets
Business innovation and optimization services
Provide for better decision making with real-time business information
Interaction services
Enables collaboration between people, processes and information
Process services
Orchestrate and automate business processes
Information services
Manages diverse data and content in a unified manner
IT service management
Manage and secure services, applications and resources
ESB Enable interconnectivity between services
Partner services
Connect with trading partners
Business app services
Build on a robust, scalable, and secure services environment
Access services
Facilitate interactions with existing information and application assets
Infrastructure services
Optimizes throughout availability and performance

Figure 2. The middleware view of the SOA reference architecture (After Flurry, 2008)

d. Layer 4 (Business Process)

Compositions of services showcased in layer three are outlined in this level. Users utilize service composition to associate clusters of services into flows, or to certain extend services are choreographed into flows; applications are then established out of services. The applications support distinct use cases and business developments (Rob, Kinder, & Graham, 2005). For this to happen, visual flow composition utilities are used for designing application flows. Figure 3 indicates how a business development P can be executed by means of services A and B, C and D from service layer. The development P comprises of the logic for the order in which services are required to be raised and executed. Services that are summed up as a business development, or flow, can be composite services or individual services constituted of distinct services.

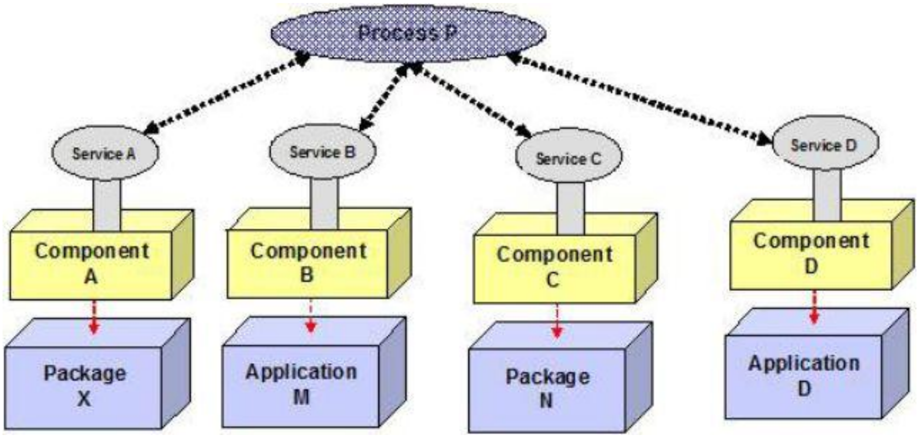


Figure 3. Services orchestration (After, The Open Group, 2013)

The business development layer shields the process representation, building blocks and composition methods for summing up loosely attached services as a chronological succession process aligned to business objectives. Control flow and data flow are utilized to aid interactions between business developments and services. The interaction may be within a single business entity or across multiple business ventures. This layer is constituted of information exchange flow between contestants (single users and business ventures), resources and processes in an array of forms to achieve desired business objective. Utmost exchanged information may also comprise of no transactional and nonstructural messages. Business logic is applied to form service flows such as parallel projects or sequential projects centered on business guidelines, policies and other business necessities. The layer also has information concerning data flows within a single enterprise or across several enterprises.

e. Layer 5 (Consumers)

The consumer layer (also known as Presentation Layer) offers the capabilities required to convey IT functions and information to end users who meet particular usage preferences. The consumer layer also provides a medium for application-to-application communication (Rob, Kinder, & Graham, 2005). Within SOA solution stack, the consumer layer offers the capability to rapidly create the front end of business procedures and composite applications. These attributes respond to differences in user demands through channels, rich clients, portals and other relevant mechanisms. It facilitates channel-independent access to particular business processes held by several platforms and applications. It is of the essence to note that SOA dissociates the user interface from the modules. The consumer layer provides SOA with a medium of integration between the underlying SOA and consumer requests. It alienates dependencies from how services are executed and who the consumers are. The architecture sets a platform where industries and organizations maintain consistent quality standards and common implementations.

f. Layer 6 (Integration)

The integration layer is considered the key enabler for SOA because it has the proficiency to mediate, course and deliver service prompts from the service client to the intended service provider. The integration layer introduces reliable set of capabilities. Integration layer has plug to plug capabilities for firm attachment of endpoint combination as well as powerful intelligent routing, protocol mediation and additional transformation mechanisms frequently provided by enterprise service bus (ESB). Web Services Description Language (WSDL) stipulates a binding, which infers the position where a service is delivered. An ESB, on the contrary, provides a location self-regulating properties for integration (Graham, 2004).

The type of integration that emerges here is predominantly the integration of layers two through four. This is the typical layer that offers communications, request, and worth services between contiguous layers in an SOA. As shown Figure 4, the integration layer delivers a cadre of indirection amid the user of functionality and its respective provider. A service user communicates with the service provider via the integration layer. Consequently, each service description is only showcased through the integration layer that is never direct for instance, an ESB and WMB. This layer also functions to decouple consumers and providers, permitting for integration of dissimilar systems into new solutions.

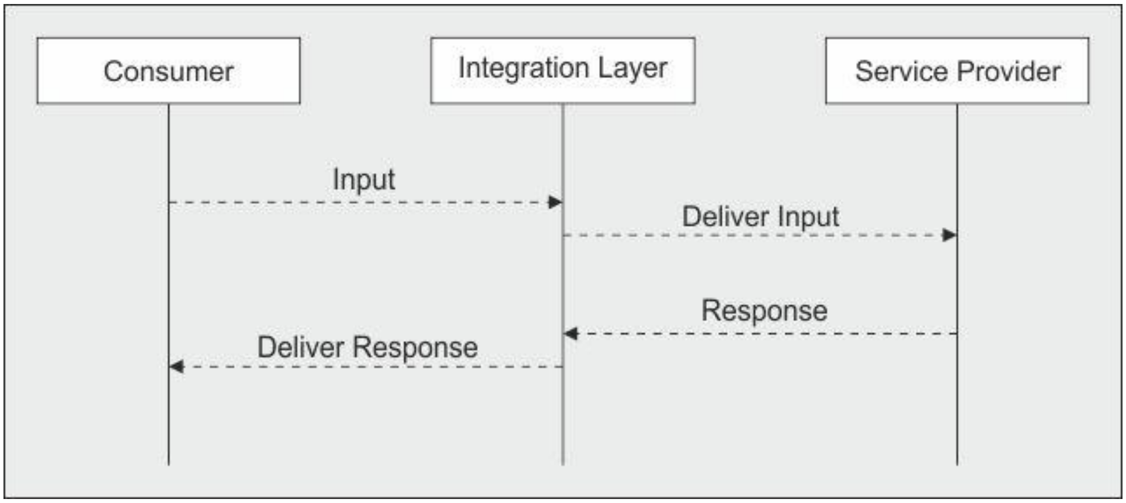


Figure 4. Interaction diagram of the integration layer

g. Layer 7 (Quality of Service)

Inherent to SOA are features that degrade existing QoS issues in computer systems. Among the features are; loose coupling, inflated virtualization, extensive use of XML, composition of federated services, decentralized SLAs, heterogeneous computing infrastructures and the requirement to sum up IT QoS metrics to yield business metrics (Lessanu, 2012). These features create difficulties for quality of service that evidently require attention within any of the SOA solution.

The QoS layer offers SOA with the capabilities needed to recognize nonfunctional requirements (NFRs). It must also enumerate, monitor, log, and indicate noncompliance with the requests linking to the pertinent service values allied with every SOA layer. This layer functions as a monitor to the rest of the layers and can release signals or proceedings when a noncompliance situation is detected or, rather, in the event that noncompliance condition is foreseen. Layer 7 creates non-functional demand related issues as a principal feature or interest of SOA and offers a focal point for carrying on with them in any available solution. This layer renders the means of guaranteeing that SOA meets its demands with respect to reliability, adequacy, manageability, scalability, and safety. Finally, it heightens the business worth of SOA through supporting businesses to enumerate the business developments contained in SOA with reference to the business Key Performance Indicators (KPI) that they impact.

h. Layer 8 (Information Architecture)

The business intelligence layer and information architecture safeguards the inclusion of vital considerations regarding data architecture and information architecture that are also applicable as the basis for the establishment of business intelligence via data marts and data warehouses. This comprises of metadata content, which is warehoused in this layer, and also the business intelligence considerations as well as information architecture.

Much applicable to industry-particular SOA assistance, the information architecture layer covers cross industry plus specific data structures, XML schema (XML-based metadata architectures) and business protocols for interchanging business data. Selected discovery, data analytic modeling and data mining are captured in this layer (The Open Group, 2013).

i. Layer 9 (Governance)

The governance layer captures all the attributes of business operational growth controlling in SOA. It prescribes direction and policies for decision-making about SOA and handling all features of SOA solution including: performance, capacity, monitoring and security. This layer facilitates SOA governance servings to be completely integrated by stressing the operational development management attribute of SOA. This layer functions concurrently with other layers in the SOA solution stack. Governance layer assist to implement QoS and make suitable application for performance metrics. It is perfectly connected with the seventh layer.

This layer can accelerate the SOA solution scheduling and design process. This layer delivers a flexible and extensible SOA governance outline that comprises of solution-level, service level pacts based on KPI and QoS, a package of performance management and capacity planning strategies that design and tune-up SOA solutions as well as solution –level security facilitation procedures from a federated applications viewpoint (Microsoft, Inc., 2006). The architectural choices in the governance layer is encrypted in consulting practices, architectural artifacts, frameworks, records of SOA capacity scheduling, SOA performance monitoring procedures, any SOA- solution SLAs and SOA solution-level security enforcement plans.

3. SOA Quality Attributes Descriptions

The key to succeeding with SOA is in comprehending the meaning and significance of its most fundamental building block: the service attributes. It is through an understanding of service attributes that truly “service-oriented” solution logic can be created in support of achieving the strategic goals associated with SOA. The primary goals of SOA are to enable analysts to access the right information at the right time and to effectively inform or make decisions. To a large degree, SOA is really providing information on demand. The description of the service attributes are listed in Table 2.

Table 2. SOA Attributes

Attribute	Rationale
Scalability	
It is realized by allocating services across various components with each component attending to a single focus for instance: validation service, identifier and user management.	➢ The capacity to support numerous components or interactions between components with a dynamic configuration
Reliability	
It is the amount of time the system takes to boot and operate effectively.	➢ It is used to minimize time between failures.
Configurability	
Configuration Management is the act of naming, changing control, automating and managing IT resources and assets	➢ SOA solution stack imposes unique needs for configuration management. ➢ A number of traditional configuration management tools and procedures are easily implemented into SOA practice.
Testability	
It is the extent of difficultness in which software can be manipulated to portray its faults.	➢ SOA utilizes this attribute to improve regression testing efficiency especially on the frequently changing business services ➢ This SOA attribute tests the overall application including the independent reusable services which are often bypassed by other architectures ➢ Frequent and improved testing implies existence of fewer defects and better general level of quality
Interoperability	
This attribute simply refers to the ability of sharing data. Highly interoperable software programs have higher chances of sharing information	➢ Software programs that are least interoperable must be integrated ➢ One of the aims of SOA is to institute interoperability amongst services for the purposes of reducing integration
Availability	
This is the extent to which a component or a system is functional and is accessible on demand.	➢ SOA initiates availability of services from both the service provider and the user's perspective. By this SOA reduces the possibility of dire consequences if one of the services becomes unavailable
Usability	
It is a degree at which the quality of user experience is determined through interaction with the services or information	➢ It initiates a more usable system
Security	
Security is confidentiality, integrity, authenticity and availability of information	➢ SOA provides security to information though a heightened security level leads to slow performance.
Performance	
This is the period that takes the system to process a request. It also determines how many requests can be processed at a specific unit of time	➢ Essential for meeting deadlines ➢ Performance marries well with the quality attribute of SOA.

4. Benefits of SOA

SOA adequately supports the problematic issue of dealing with constantly changing technologies and also supports integrating disparate systems and applications that are built using different technologies and infrastructures, which hamper interoperability and seamless integration. This solution provides a powerful abstraction which identifies all compute resources as entities that can be dynamically discovered and composed. These entities referred to as services (Layer 2): are described in terms of interfaces specifying service functionality independent of platform technology or programming language used. This renders the service abstraction particularly advantageous when applied for tackling problems due to heterogeneity of IT landscapes. The concept of SOA is the bridge between interoperability goals and the set shortfalls introduced in Section A of this chapter.

It is important to reexamine how a SOA solution supports poor interoperability issues. First and foremost, SOA aids organizations transform their business processes to high performance by simplifying the interfaces between existing information systems with newer technologies. SOA enables organizations to respond quickly to new business requirements, develop unique new capabilities and leverage existing services for true responsiveness making IT systems more closely aligned with each other. SOA promotes the reuse of existing assets, increasing efficiency and reducing application development costs. It also enables IT systems to quickly leverage the most readily available code bases and services from across any organization. Furthermore, they improve coordination across the entire organization in order to reduce time-consuming problem resolution.

SOA also allows organizations to meet their standardization IT goals. The technological values of SOA are based on industry standards and can decrease complexity when compared with integrating systems on a non-standardized basis. They also enable future applications to network seamlessly with existing standards-based services. SOA allows simplicity and ease of maintenance reducing support costs and freeing up IT staff for strategic work. In addition, connectivity, data exchange and process integration efforts are simplified, reducing integration-related development and support costs. SOA represents software assets as services and provides a standard way of representing and interacting with software assets.

Finally, SOA solutions address heterogeneous systems by providing an enterprise-level view of services, and offer the ability to decrease time to implement the enterprise resource planning (ERP) solution, while reducing IT resource exposure through service reuse. More importantly, the design of SOA solutions with a business focus ensures the relevancy and the value of technology to the organization.

5. Impact of SOA

This chapter addressed the issue of interoperability and gives an overview of the most important aspects of SOA from the point of view of industry best practices, Marine Corps exercises and academia. The overall goal is the provision of SOA model, whereas the major benefit of services is revealed by its flexibility in reuse and considerably easier integration effort. SOA objectives can be summarized as the following; (1) determine which services or partial service are possible for interoperability solutions; (2) demonstrate SOA model of interoperability supported by best practices and; (3) identify techniques in which SOA can contribute solutions to the interoperability problem. Based on the critical LCM systems' diversification and interoperability problems to solve, this solution addresses these challenges by systemizing disparate systems and is highly dependent on standardization which enables reuse of legacy applications with newer technologies. SOA goals aim at solving integration problems by improving efficiency and effectiveness throughout the overall IT architecture in order to provide accurate and timely data for superior decision-making.

THIS PAGE INTENTIONALLY LEFT BLANK

III. THE BUSINESS PROCESS

A. MLS2 CURRENT PROCESS

1. Business Process Modeling

This chapter uses a Business Process Modeling (BPM) tool known as Savvion. Savvion is a BPM product that provides modeling, documentation, automation, optimization and monitoring of processes across a wide set of systems (Hailstone, 2009). A comprehensive BPM tool, such as Savvion, provides the ability to collectively define an organization's business processes. The advantages of using BPM tools are that processes can be integrated with existing software systems, decision-makers have near real-time visibility they need to monitor, analyze, control and improve the execution of those processes which increases operational responsiveness. BPM compliments SOA because it incorporates business rules and processes with existing operational systems such as MLS2 and legacy systems. In addition to business process management technology, BPM tools provide solutions for business event processing and transaction assurance which facilitate data interoperability.

BPM is an important step towards an SOA solution because it defines and outlines business practices, processes, information flows, data stores and the IT architecture used for these major processes and work flows. It is a holistic management approach to aligning an organization's business processes with the wants and needs of clients (vom Brocke, 2010). It supports business efficiency and effectiveness while undertaking innovation, flexibility, and integration with technology. It enables organizations to be more efficient, more effective and more capable of change than a functionally focused, traditional hierarchical management approach (Ko 2009). BPM is supported and enabled through technology to ensure the sustainability of the managerial approach in times of change.

SOA and BPM are a perfect complement to each other because they provision interfaces across functions that are often hampered by a lack of interoperability of disparate underlying systems. BPM and SOA expose areas where processes can integrate with IT. Implementing business processes on a BPM and SOA foundation means the business services are executed as business transactions flow through the process. By placing probes on these business services to collect service performance and other metrics, organizations can gain real-time visibility into their business that otherwise would be hard to achieve.

In the drive for interoperability and agility, BPM is based on the principles of SOA. Both aim for faster response to changing business requirements, including compliance, mergers and acquisitions, and product and service introductions. SOA architecture has become a crucial foundation for BPM, supporting rapid assembly and orchestration of process services into larger, end-to-end processes. BPM based on SOA offers an environment that changes the traditional process for altering an application to reflect changed business rules or processes. It places the controls for change management in the hands of the business process owner rather than on IT's shoulders. Through intuitive, visual interfaces, effective BPM environments offer managers ways to change rules and alter processes without having to drop down to the coding level. The objective of BPM is to interpret core processes with technology capabilities in order to mutually support one another through a sharing of information and data exchange. This chapter will introduce MLS2's current procedures via Savvion to examine MLS2's TLOC systems and their current processes in order to determine where IT integration can be implemented.

2. MLS2 Visio Flow Chart

The Visio business flow diagram (Figure 5), demonstrates the existing process that is utilized in TLOC software systems for the request, receipt, processing, tasking and tracking of logistics support within the MEFs. These core logistic processes measure valuable metrics such as order and ship times, repair cycle times and overall logistic response times. It is important to define each logistics process and sub-process in order to adequately measure and improve the procedures and systems that define response metrics (Robbins, Boren, Eden & Relles, 1998). These are some of the systems currently employed at the TLOC throughout MAGTF's LCEs:

- The Common Logistics Command and Control System (CLC2S) is a tactical web-enabled logistics information management system designed to provide Marine Air Ground Task Force (MAGTF) with enhanced capabilities to assess, plan, and execute logistics functions to achieve mission objectives. CLC2S can provide near real-time asset visibility, asset management capabilities, decision support tool sets, and integrated request management in a distributed, rapidly changing battlefield environment. The system has been designed to be highly configurable, to operate on the Marine Corps tactical communications infrastructure and to aggregate logistics data by means of integration with legacy data systems.
- Transportation Capacity Planning Tool (TCPT) is a net centric/web accessible tool that aids with the planning, tracking, management, and execution of transportation centric missions. TCPT provides transportation and logistics commanders with transportation capacity planning via a digital dashboard view of all available transportation assets, mission requirements, and essential elements of information to aid with executing his current and future transportation missions.
- Battle Command Sustainment Support System (BCS3) is a map-centric display on a commercial laptop that provides a technical and visual picture of the battlefield. BCS3 allows In-Transit-Visibility (ITV) to be graphically displayed on the common operating Picture (COP) accessible across the entire supply chain in order to enhance decision-making abilities and better support operationally-deployed units.
- Supported Activities Supply System **(**SASSY) is the legacy intermediate level supply system. SASSY is the HQMC mandated record keeping control and data collection agency.
- GCSS-MC is the primary ERP technology enabler for the Marine Corps Logistics Modernization strategy and provides the backbone for all logistics information required by the MAGTF. The core is modern, commercial-off-the-shelf enterprise resource planning software (Oracle 11i e-Business Suite). GCSS-MC does not currently provide capabilities to the warfighter while deployed or an all-inclusive solution to all functions of logistics (these capabilities will be released in future increments).

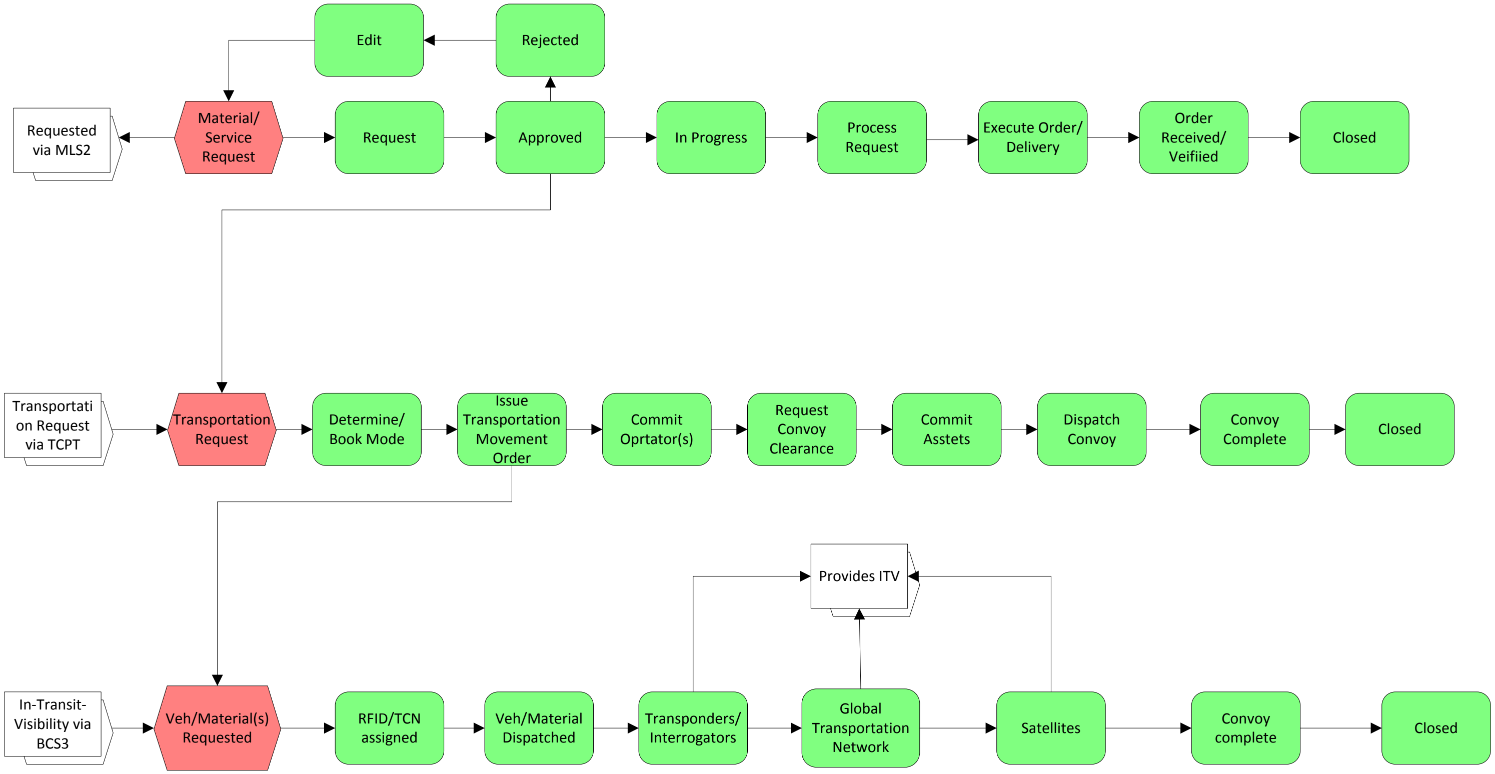


Figure 5. MLS2 Process Flow Chart

With the exception of a limited point-to-point interface between MLS2 and TCPT, these systems are considered stand-alone commercial-of-the-shelf systems (COTS). COTS are normally a prebuilt software solution supplied to the government by a vendor via an identified systems' requirement (Morisio, et al., 2002). Due to limited resources to build and implement an ERP solution, COTS solutions are intended at meeting an interim solution for single requirements with the notion to incrementally work towards an ERP result. The problem is that each COTS solution is often given to a sole vendor causing even more fragmentation between systems due to distributed software support from various vendors. Therefore, most COTS products only add to the integration issues and additionally introduce a dependence on countless vendors for software support.

a. Current Process: Mission Impact

The current process emphasizes an urgent need for improving the Marine Corps re-supply procedure. The compelling need to make a radical change also underlies in the Marine Corps' current supply system known as Supported Activity Supply System (SASSY). SASSY was created in the 1970s and was designed for inventory control, accountability, requisitioning of supplies and management of fiscal data. Aside from its antiquated state, SASSY is difficult to learn and presents inaccurate, untimely data. SASSY is a stove-piped system that does not interface with other intermediate/wholesale supporting systems, and therefore the transfer of data between this mandated legacy system and the MLS2s is either point-to-point or nonexistent.

This lack of interoperability between these mutual supporting systems may cause a unit outside the United States in a forward position, which cannot internally support itself, to wait for parts from back in the States. The unit may be collocated near another supply depot but SASSY "never knows" because there is no interface between supporting systems in their area of operation. The SASSY customers have become unsympathetic due to unfulfilled promises from SASSY leading to "no faith" in the system. The current speed of information flow, time, money, and resources are more crucial now than ever, there is no time to wonder when or if re-supply will occur. Units need to have faith that they can place an order once and it will arrive in a timely manner. Large gaps also exist in the lack of total asset & in-transit visibility information which is facilitated via BCS3. The lack of visibility on unit stocks and in-transit visibility on ordered items makes it difficult to identify actual shortages, to locate needed items with in stocks for reallocation, and to direct and track the movement of ordered items to requesting units. A universal, more timely and accurate supply system is required to keep up with the operational tempo on the ever-evolving technology.

The effect on capabilities for logistics to perform its mission if interoperability between systems is not provided includes;

- The Marine Corps Logistics Chain will continue to operate in a disjointed, segmented, and stove-piped method with multiple systems that do not interface. Data will remain untimely, inaccurate and provide no ITV, TAV or situational awareness of functional logistics chain management.
- LCE Commanders will continue to manually determine capacity and capabilities; dedicating time, personnel and resources rather than leveraging available technology solutions.
- LCE Commanders will lack automated tools in order to assist with planning, estimating, tasking, monitoring execution and better decision-making techniques.

3. Savvion (As-Is Model/Metrics)

Organizations are dependent on the successful execution of their operational processes that control core functional areas (Hailstone, 2009). Savvion gives us a formal method to understand processes and identifies potential inefficiencies and bottlenecks in order to provide solutions for more efficient and effective process flows. The approach taken with Savvion helped identify improvements in TLOC software systems by simulating current processes and helping to determine required resources to avoid the bottlenecks for the request, receipt, processing, tasking and tracking of logistics support.

a. The Process

MSCs typically submit a daily courier (parts request) to their perspective Combat Logistics Battalion (CLB); CLB manages Class IX (repair parts), secondary repairable, and miscellaneous parts.

- Couriers are submitted daily to the CLB by each MSC (via SASSY and/or CLC2S)

Once a courier is submitted it is cycled through the organic account via ATLASS/SASSY in order to check against the Class IX on hand quantities. The request is either filled or passed to the supporting unit (SMU is the intermediate supporting unit in CONUS). The "pass" process is automatically cycled through ATLASS/SASSY. In addition, an offline request is sent to TCPT for transportation support.

- If the CLB has item O/H; it is pulled from the inventory and released to the customer.
- If the CLB does not have item O/H; it is passed to supporting SMU.
- CLC2S submits request to TCPT at local TLOC (offline process).

Once a part request is received and cycled through the SMU account via SASSY and CLC2S, it checks the courier against the on-hand quantities. The request is either filled or placed on back-order to the alternate source of supplier (SOS). Once item is placed on back-order, it is loaded on the SMU's general account balance file (GABF) until available.

- If the SMU has item O/H; it is pulled from the inventory and released to TCPT for delivery to the customer.
- If the SMU does not have item O/H; it is placed on back-order and passed to alternate Source of Supply (SOS).

As previously mentioned, MLS2 systems are COTS products. Therefore, TCPT's limited connectivity to CLC2S obligates the users to make manual updates between status updates (i.e., vehicle and personnel availability and in-transit-visibility TLOC updates to establish common operating picture). When TCPT dispatches vehicles, the in-transit-visibility tracking updates are made via BCS3. This is problematic because there are multiple competing system transactions to fill one request.

b. Savvion Model Data Input Assumptions:

The following items listed below were inputs into Savvion which facilitated the scenario and data capture;

- Requisition has already been approved at the MSC funding level before process begins (funding actions were not included in model).

- All requests have equal priority (routine request); outside routine request involve alternate systems outside of thesis scope.
- The Material Release Order (MRO) is inclusive of time it takes to pick, pack and deliver the requisition. This is also considered the average customer wait time or order ship time.
- Work days: Deployed unit workdays are 12 hours; CONUS workdays are 8 hours.
- Requisition Fill Rates %s used in our Savvion Model are average estimates from SMU's historical data percentages.
- Numbers of instances to number of intervals between instances are based on 7 work days in a week for duration of 60 days. (500 instances at 90 minute intervals)

Figure 6, depicts the current As-Is process via Savion model.

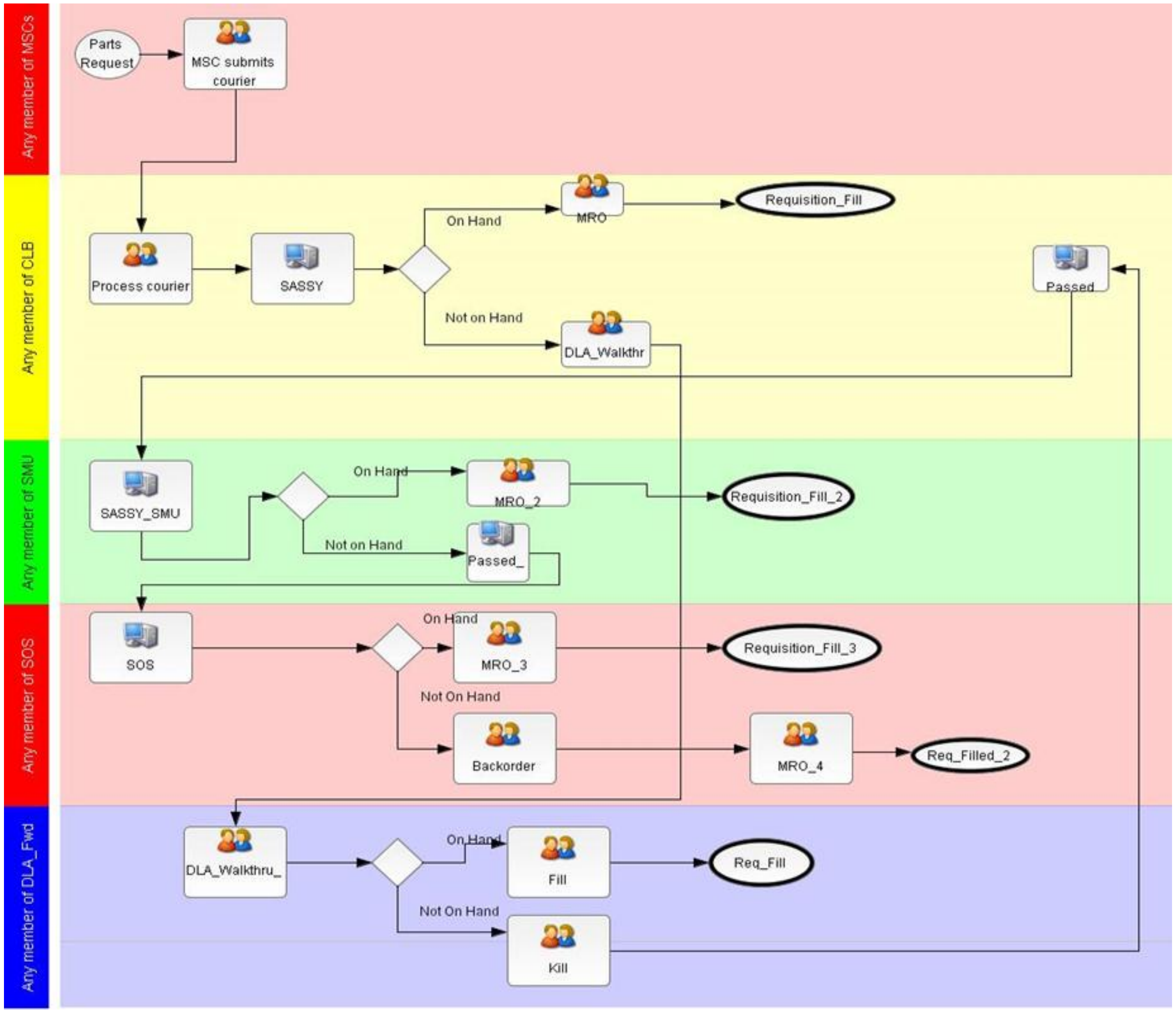
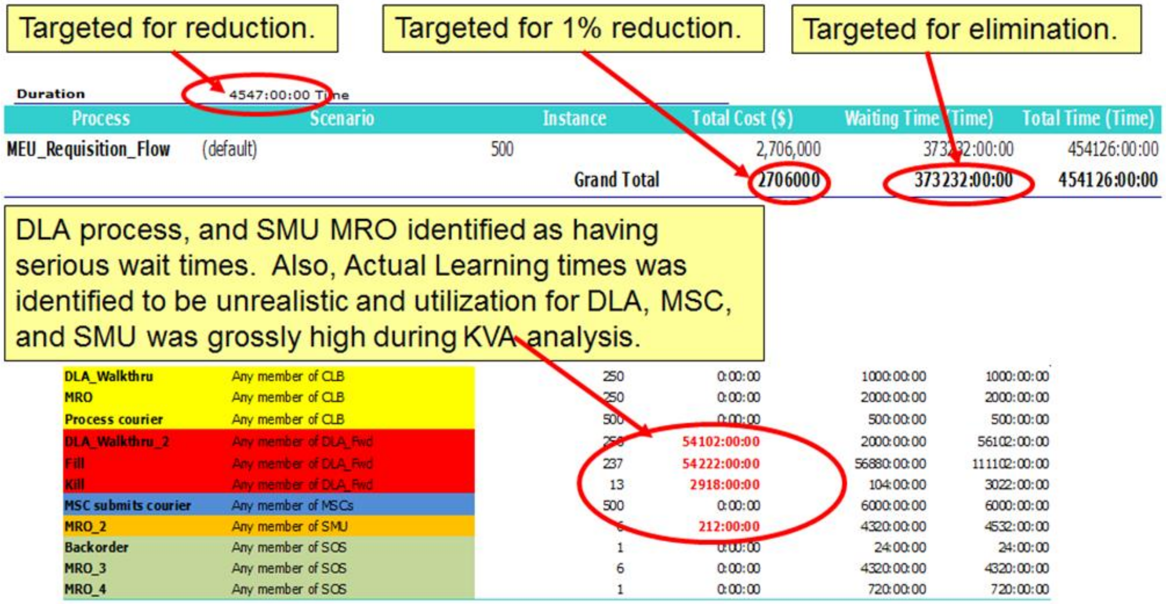


Figure 6. As-Is Savvion Model

c. Savvion Results and Issues with Current Process

The current process lacks the ability to create situational awareness of functional logistics capabilities and capacities. The metrics (Figure 7) and listed bullets highlight the current process issues identified via Savvion;



Targeted for reduction.

Targeted for 1% reduction.

Targeted for elimination.

Duration 4547:00:00 Time

Process	Scenario	Instance	Total Cost ($)	Waiting Time (Time)	Total Time (Time)
MEU_Requisition_Flow	(default)	500	2,706,000	373232:00:00	454126:00:00
		Grand Total	2706000	373232:00:00	454126:00:00

DLA process, and SMU MRO identified as having serious wait times. Also, Actual Learning times was identified to be unrealistic and utilization for DLA, MSC, and SMU was grossly high during KVA analysis.

DLA_Walkthru	Any member of CLB	250	0:00:00	1000:00:00	1000:00:00
MRO	Any member of CLB	250	0:00:00	2000:00:00	2000:00:00
Process courier	Any member of CLB	500	0:00:00	500:00:00	500:00:00
DLA_Walkthru_2	Any member of DLA_Fwd	250	54102:00:00	2000:00:00	56102:00:00
Fill	Any member of DLA_Fwd	237	54222:00:00	56880:00:00	111102:00:00
Kill	Any member of DLA_Fwd	13	2918:00:00	104:00:00	3022:00:00
MSC submits courier	Any member of MSCs	500	0:00:00	6000:00:00	6000:00:00
MRO_2	Any member of SMU	6	212:00:00	4320:00:00	4532:00:00
Backorder	Any member of SOS	1	0:00:00	24:00:00	24:00:00
MRO_3	Any member of SOS	6	0:00:00	4320:00:00	4320:00:00
MRO_4	Any member of SOS	1	0:00:00	720:00:00	720:00:00

Figure 7. Savvion As-Is Metrics

1. Lack of faith in current process lends to hoarding and over ordering actual quantities required. It also generates bottlenecks creating long duration (order-ship times) in the system by adding too many requests for a part which may holdup production, distribution, and reporting units.

2. SASSY is not a real time system which adds o duration and also lacks interoperability with other systems. Couriers submitted via SASSY are batch uploaded daily and cycled overnight therefore updates to the status of your request are not made available until the next day. Additionally, there are no legacy system feeds with MLS2 technologies.

3. There are deficiencies with total asset visibility (TAV) and in-transit-visibility (ITV). The absence of TAV on unit and/or local stocks excludes abilities to locate

required items with in readily available stocks. The lack of ITV on ordered items makes it challenging to direct and track the movement of ordered items to requesting units.

4. There are multiple manual updates throughout the requisition process. These laborious procedures cause bottlenecks in the system creating backlogs and extended periods of wait times.

d. Savvion Reengineering Goals

Several factors went into ensuring the success of this business process reengineering scenario. The listed details were the goals set to reengineer the As-Is process in Savvion:

1. Reduce duration of parts requisition customer wait time by eliminating stove-piped system process; streamline process and exclude dual processes where systems lack interoperability.
2. Data exchange/interfaces between MLS2s and legacy systems. Recommend that MLS2s provide a COP via an integrated network. Provide courses of action for mutual support and architecture views between MLS2s and legacy systems.
3. Improve effectiveness by eliminating manual processes in order to project better logistics planning and estimations.
4. Provide ITV and TAV information that is accessible across the entire supply chain in order to enhance decision-making abilities and better support operationally deployed units.
5. Improve efficiency by providing enterprise level data for all units and commands throughout the Marine Corps.

4. Savvion (To-Be Model/Metrics)

The To-Be Savvion metrics were directly associated to linking business case data with calculated goals and measurable objectives. An end-to-end perspective was taken into account for the entire process, and it involved multiple stakeholders of the organization that played roles in elements of its execution. Figure 8 depicts changes made to requisition process in order to meet reengineering goals.

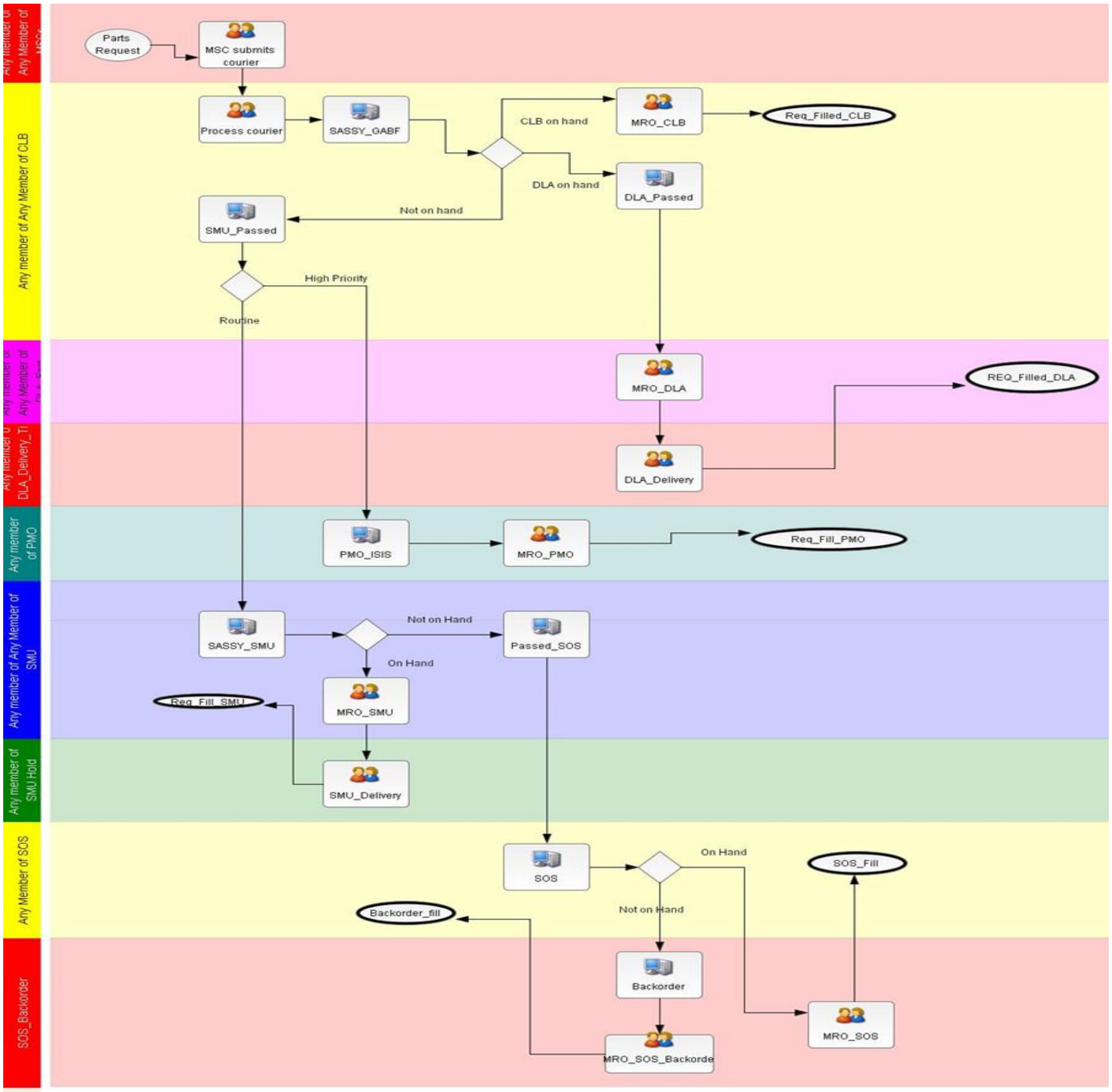


Figure 8. To-Be SAVVION Model

a. Savvion Process Revision Results

Duration: Overall duration was reduced from 189 to 60 days. The following were the specific reengineering objectives that were made in order to reduce duration:

- Automated the manual and dual processes.
- Added Priority Management Office (PMO) into the process. PMO is a Naval Logistic Initiative that supports deployed units with high priority requisitions and eliminated the gap with BCS3. This office utilizes commercial distributors in order to expedite mission critical parts. PMO helped reduce duration in the process by eliminating the passes that would have been sent back to SMU CONUS. Wait times from SMU are an average of 30 days and whereas PMO delivers an MRO within 5 working days. Aside from reducing overall wait time, the PMO process added value by reducing the lack of ITV and TAV therefore decreasing down time of mission essential equipment that the Commander requested in order to accomplish the unit’s mission(s).

Cost: Cost was reduced to $76K for 500 requisitions which is ~$150 to process each transaction. By interfacing and streamlining processes, the reduction efforts brought cost down by over 90% from original processing rate. The following were the specific reengineering objectives that were made in order to reduce cost:

- Number of personnel were reduced to an overall cost savings of 56% which saved ~$140 an hour. MSC members were reduced from 12 to 8 which resulted in a 60% savings ($53/hr savings). CLB members were reduced from 10 to 5 which resulted in a 45% savings ($60/hr savings). SMU members were reduced from 7 to 5 which resulted in a 64% savings ($26/hr savings).
- Process improvements in reducing overall duration attributed to 34% of the cost savings. By eliminating wait times and automating manual processes, the Savvion simulated work times were reduced or completely eliminated. For example, the expeditors that supported the ITV process were removed from the model once the automation replaced that manual process. Reductions in work times were implemented for MRO and Pass processes which also contributed to the overall cost reduction.

b. Savvion Model Conclusions

Because SOA and BPM overlap each other in terms of what they seek to accomplish, both concepts are in many ways inseparable. While one could imagine BPM as a more logical design approach, its principles are firmly rooted in optimizing business technologies. BPM applied to SOA covers process alignment and provides building blocks for aggregating loosely-coupled services as a sequencing process aligned with business goals. Data flow and control flow are used to enable interactions between services and business processes. The interaction may exist within an enterprise or across multiple enterprises. By mapping out processes via BPM, the outcomes in figure 9 can be used to drive a detailed interface between particular actions that trigger information exchange in order to facilitate an interoperable process.

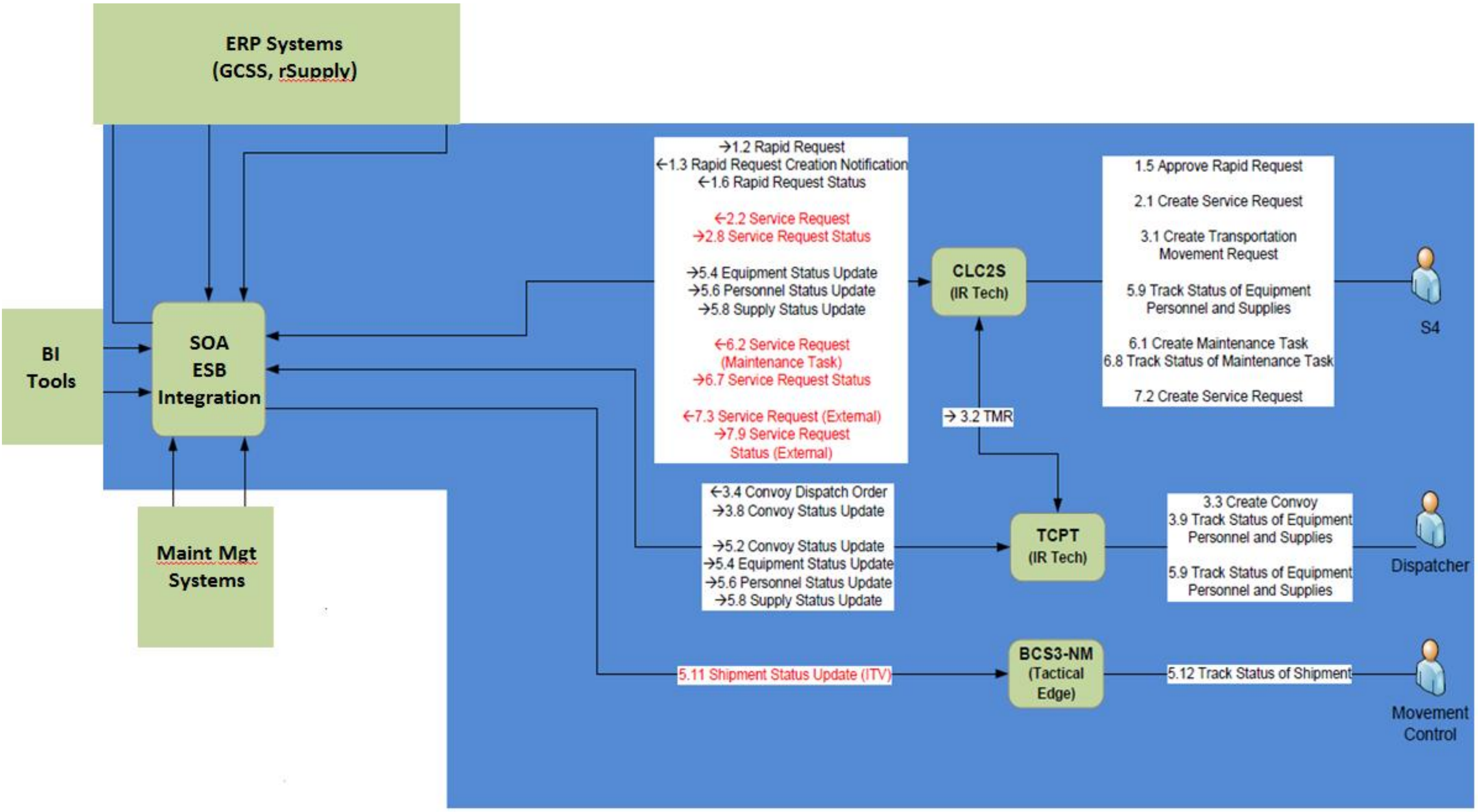
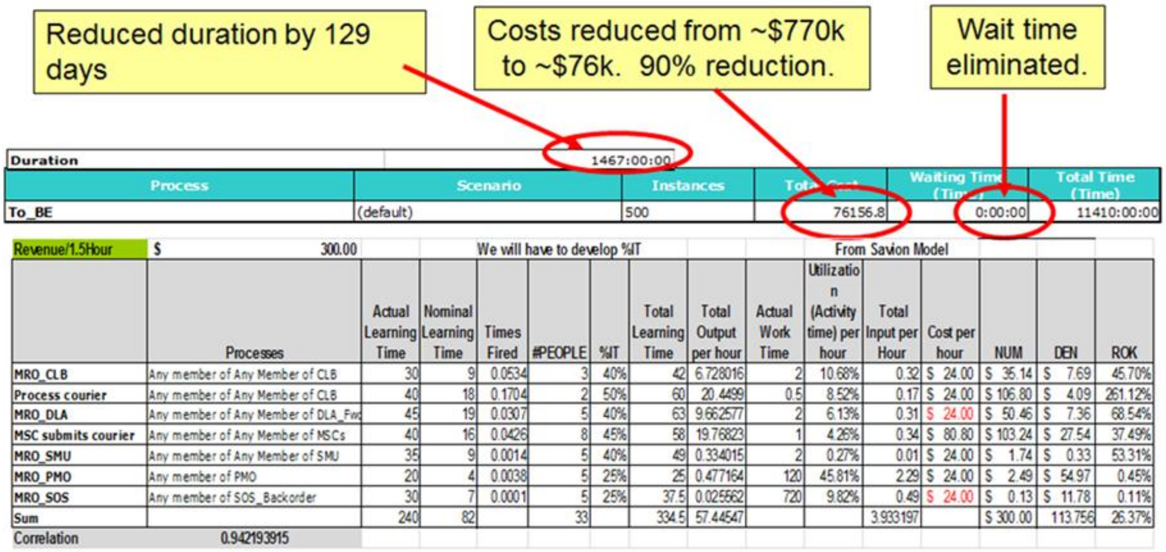

ERP Systems
(GCSS, rSupply)
BI
Tools
SOA
ESB
Integration
Maint Mgt
Systems
→1.2 Rapid Request
←1.3 Rapid Request Creation Notification
←1.6 Rapid Request Status
←2.2 Service Request
→2.8 Service Request Status
→5.4 Equipment Status Update
→5.6 Personnel Status Update
→5.8 Supply Status Update
←6.2 Service Request
(Maintenance Task)
→6.7 Service Request Status
←7.3 Service Request (External)
→7.9 Service Request
Status (External)
CLC2S
(IR Tech)
1.5 Approve Rapid Request
2.1 Create Service Request
3.1 Create Transportation
Movement Request
5.9 Track Status of Equipment
Personnel and Supplies
6.1 Create Maintenance Task
6.8 Track Status of Maintenance Task
7.2 Create Service Request
S4
→ 3.2 TMR
←3.4 Convoy Dispatch Order
→3.8 Convoy Status Update
→5.2 Convoy Status Update
→5.4 Equipment Status Update
→5.6 Personnel Status Update
→5.8 Supply Status Update
TCPT
(IR Tech)
3.3 Create Convoy
3.9 Track Status of Equipment
Personnel and Supplies
5.9 Track Status of Equipment
Personnel and Supplies
Dispatcher
5.11 Shipment Status Update (ITV)
BCS3-NM
(Tactical
Edge)
5.12 Track Status of Shipment
Movement
Control

Figure 9. MLS2 SOA Key Alert Objects

Additionally, the Savvion BPR reengineered adjustments to the process added the following value; the metrics shown in Figure 10 highlight the Savvion process revision results:

- Lowered costs by 90%
  - By reducing manpower, 56% of overall cost was cut
  - Automation of manual processes saved 34% of cost
- Reduced total number of requisitions to CONUS (SMU)
  - Decreased overall order ship time or duration in Savvion
  - Increase in Material Unit Readiness; this metric was not captured in Savvion but the process expedited repair parts which attributes to equipment readiness to the Commander


Reduced duration by 129 days
Costs reduced from ~$770k to ~$76k. 90% reduction.
Wait time eliminated.

Duration			1467:00:00		
Process	**Scenario**	**Instances**	**Total Cost**	**Waiting Time (Time)**	**Total Time (Time)**
To_BE	(default)	500	76156.8	0:00:00	11410:00:00

Revenue/1.5Hour	$ 300.00			We will have to develop %IT						From Savion Model					
	Processes	Actual Learning Time	Nominal Learning Time	Times Fired	#PEOPLE	%IT	Total Learning Time	Total Output per hour	Actual Work Time	Utilization (Activity time) per hour	Total Input per Hour	Cost per hour	NUM	DEN	ROK
MRO_CLB	Any member of Any Member of CLB	30	9	0.0534	3	40%	42	6.728016	2	10.68%	0.32	$ 24.00	$ 35.14	$ 7.69	45.70%
Process courier	Any member of Any Member of CLB	40	18	0.1704	2	50%	60	20.4499	0.5	8.52%	0.17	$ 24.00	$ 106.80	$ 4.09	261.12%
MRO_DLA	Any member of Any Member of DLA_Fwd	45	19	0.0307	5	40%	63	9.662577	2	6.13%	0.31	$ 24.00	$ 50.46	$ 7.36	68.54%
MSC submits courier	Any member of Any Member of MSCs	40	16	0.0426	8	45%	58	19.76823	1	4.26%	0.34	$ 80.80	$ 103.24	$ 27.54	37.49%
MRO_SMU	Any member of Any Member of SMU	35	9	0.0014	5	40%	49	0.334015	2	0.27%	0.01	$ 24.00	$ 1.74	$ 0.33	53.31%
MRO_PMO	Any member of PMO	20	4	0.0038	5	25%	25	0.477164	120	45.81%	2.29	$ 24.00	$ 2.49	$ 54.97	0.45%
MRO_SOS	Any member of SOS_Backorder	30	7	0.0001	5	25%	37.5	0.025562	720	9.82%	0.49	$ 24.00	$ 0.13	$ 11.78	0.11%
Sum		240	82		33		334.5	57.44547			3.933197		$ 300.00	113.756	26.37%
Correlation	0.942193915														

Figure 10. SAVVION To-Be Metrics

5. Issues and Recommendations

CLC2S, TCPT, and BCS3 are leading TLOC systems. All add capabilities to decision making but not without numerous setbacks.

Issues:

1. Without proper policy in place from Higher HQ, units lack standardization and cannot train to all MLS2 technologies. The lack of unity of effort makes it impossible to

provide functional battalions, CLBs or Detachments with TLOC capabilities that merge efficiently into existing TLOC IT systems.

2. MAGTF logistic customers must learn multiple logistic request systems depending on which LCE/TLOC receives and processes their requirements.

3. Adoption of MLS2 technologies will not be efficient in future implantation efforts of GCSS-MC roll-out without early refinement and unity of effort. GCSS-MC Block I (current roll-out) is aimed at the replacement of SASSY and MIMMS only. The LCEs, DC I&L and Training and Education Command (TECOM) must refine these MLS2s and set a common software direction in the schoolhouse, in garrison, and in operational employment to properly prepare the way for a common TLOC operating system to be fully tested and ready for integration into GCSS-MC.

4. HHQ policy must narrowly define what TLOC operating systems are to be used within our MAGTFs. BCS3, CLC2S and TCPT are the authorized TLOC systems but these systems have redundancies, gaps and are not capable of synchronizing with each other.

a. Recommendations:

In order to accomplish standardization, concurrence across all MEFs for which TLOC operating system to adopt and employ must be gained. DC I&L must provide the guidance and direct standardization across the MAGTF for ground logistic software systems. Training units (such as TECOM) must provide training and education throughout MOS and PME education and training courses. In order to properly plan for future interfaces with GCSS-MC, current integration efforts must be made in order to reduce redundancies and stove-pipe characteristics. GCSS-MC is the planned ERP system but in the interim, we must focus on the interoperability of current legacy and new technology operating systems in order to systematize processes and facilitate both present and future IT decision making capabilities.

The success of logisticians is degraded by systems that are not interoperable, nor flexible, or do not provide appropriate information to commanders in a timely manner. Degraded operational capabilities, as well as insignificant corrective measures, results in improper systems integration. Integration of LCM systems within the Marine Corps is not an easy task. However, the Marine Corps can drastically improve LCM C2 systems integration efforts with a SOA solution which could be enacted with relatively minimal instability while improving both current and future processes. Inaction could adversely affect policy, requirements, doctrine, acquisition, and post-deployment software support of LCM systems; with indecisiveness the inability to effectively perform LCM systems integration will continue to trouble the Marine Corps. In order to meaningfully integrate C2 systems throughout the Marine Corps, there must first exist a basic philosophy and understanding of the "interoperability concept" which SOA can facilitate.

IV. THE INTEROPERABILITY SOLUTION

A. MLS2 CONCEPT OF EMPLOYMENT

The scope of this chapter is to describe an alternative architecture within SOA layers and principles applicable to MLS2s. MLS2s will employ a SOA approach by obtaining services that provide the TLOCs with the ability to connect existing legacy systems with developing technologies in order to meet operational LCM C2 requirements. The MLS2 SOA approach involves an accurate interoperable MLS2 capability to collect, process, and disseminate data within LCE systems of the MAGTF. This approach provides LCE commanders with a COP in order to conduct staff planning and perform logical decision-making. The end state is a common, scalable, service-oriented capability that is seamlessly employable throughout LCM while enhancing effectiveness and efficiency through better collaboration and a shared understanding. MLS2 SOA, goals are to:

- Provide an improved, standards-based approach to achieve information sharing.
- Increase agility through effective reuse of services and capabilities.
- Replace antiquated system interfacing techniques with a SOA-based integration methodology.

The intention of SOA is to implement MLS2 software products which provide the foundation to deliver capabilities quickly to the Marine Corps in a shared operating environment. Through a collaborative architectural structure, SOA leverages various providers to produce these capabilities. Key engineering artifacts are leveraged to decide which technologies to pursue, document why those technologies were selected, how the technologies affect the users and how the technologies are incorporated into the software architecture and design descriptions. A high level view of the software components and their integration of SOA are shown in Figure 11.

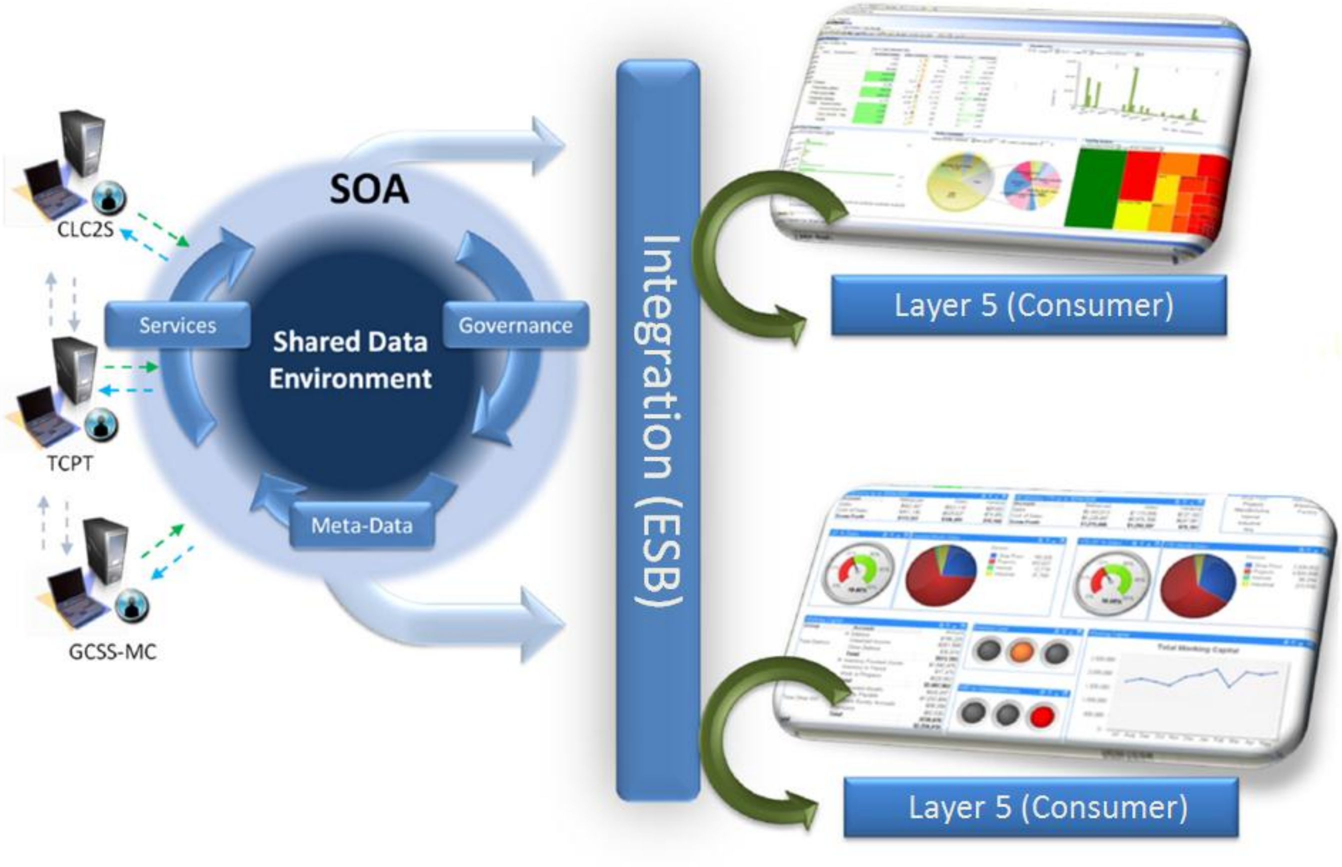


Figure 11. MLS2 Concept of Employment Overview

The ESB component provides the core asset to support a stable foundation for other software components to leverage (Malatras, 2008). By leveraging the SOA logical layers, data integration through a shred environment will enhance the speed and accuracy via this ESB interface. SOA is leveraging industry standards where possible to support interoperability internal and external to best practice software products.

B. ARCHITECTURE APPROACH

Typically each TLOC interconnects to other LCEs through Tactical Data Networks (TDNs), voice and/or Enhanced Position Locating and Reporting System (EPLRS) radios. A TLOC may interconnect to another TLOC that is physically deployed within a short distance through a direct Ethernet or serial router-to-router cable. The TLOC does not provide any TDN, Single Channel Radio, or EPLRS communications assets; these assets are determined and provided by each unit upon deployment.

It is important to point out that system configuration varies by command and allocation of resources. The TLOC, or Major Subordinate Command (MSC), could possibly incorporate majority of the MLS2 SOA elements, while the LCEs, Regiments and Battalions, will require a scaled down version of a SOA employment in order to fit within the more constrained networking environments. Additionally, a dismounted unit could involve deploying a subset of SOA components. Due to the need of deploying a variety of software configurations to the variants, it is vital that the software architecture support composability and clearly identify dependencies between software elements (Erl, 2007). This will facilitate configuration management and the ability to create software installation media that is reusable to each variant.

MLS2 SOA uses the conceptual layers, as defined in Chapter II, as a framework for describing the services within the architecture. Layers 2 through 4 from chapter II are depicted in Figure 12. Majority of data integration is covered in layers 2 through 4 which reinforce interoperability objectives.

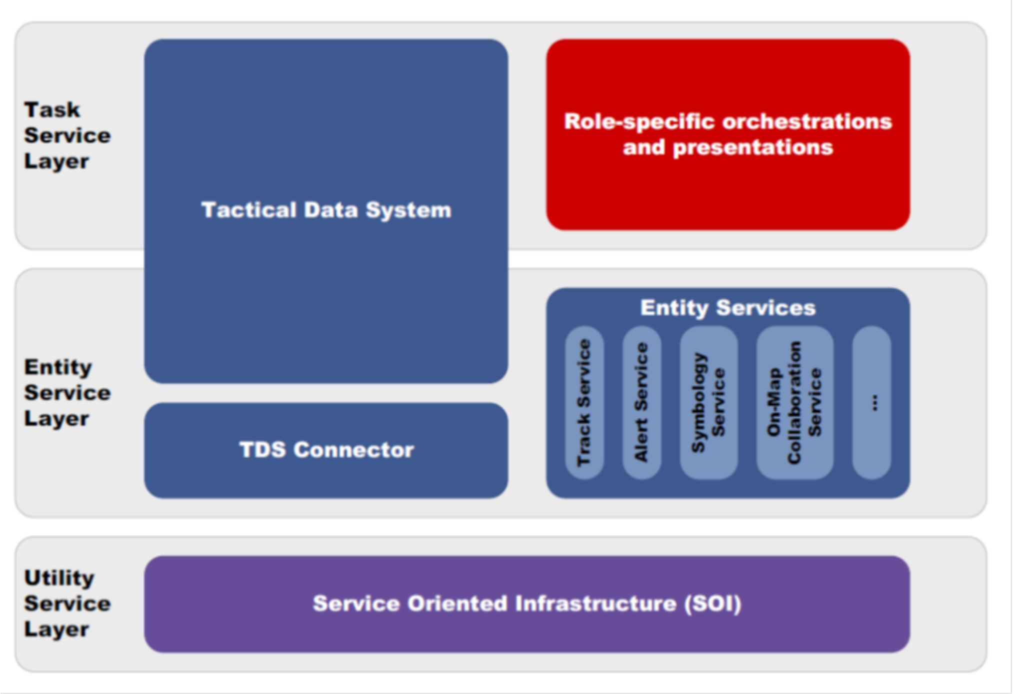


Figure 12. SOA Service Layers

- Utility Service Layer (Layer 2: Service Components) – The MLS2 SOA's Service Oriented Infrastructure (SOI) provides the non-business centric infrastructure that is the basis for all other services in the architecture.
- Entity Service Layer (Layer 3: Services) – Entity Services are primarily concerned with communicating one or more specific data types between MLS2s and the utility service layer. Entity Services focuses on specific data types such as services for tracks, alerts, traps, etc.
- Task Service Layer (Layer 4: Business Process) – As mentioned previously, most business processes are provided by the combination of MLS2 and legacy systems. This layer manages the core business compositions and performance capabilities constructed around specific operational mission threads and operator roles.

Figure 11 also supports the Marine Corp's key architectural SOA conceptual goals as intended in the Joint C2 Objective Architecture. Joint C2 IT capabilities are envisioned to provide a basis to exploit interoperability by minimizing integration risk and leveraging enterprise-based solutions (Joint, 00). These are the essential IT architectural goals for the Marine Corps:

- Implement interoperability capabilities through rapid provisioning and frequent software enhancements.

- Provide local/network connectivity and interoperability in tactical scenarios where user environments include; disconnected operations, intermittent connectivity, and limited bandwidth.
- Leverage enterprise services and information that provide interoperability in order to implement LCM C2 business processes.
- Create an infrastructure which allows several different projects to deliver functionality that provides interoperability between each other.

1. Utility Service Layer Design

This layer contains software components, each of which provides the implementation or “realization” for services and their operations. In this layer, the functional and technical components that facilitate a service component are realized in one or more services. The Utility Service Layer (SOI design) is based on the SOA composability principle in order to maximize the value of the components through service reusability and standardization (Erl, 2007). The utility service layer addresses the interoperability issue by providing a centric infrastructure and facilitating a heterogeneous network between MLS2s and legacy systems. Figure 13 illustrates a diagram for the implementation and optimizing a service and demonstrates the sequence an architect follows to provide a cohesive environment for the deployment of SOA.

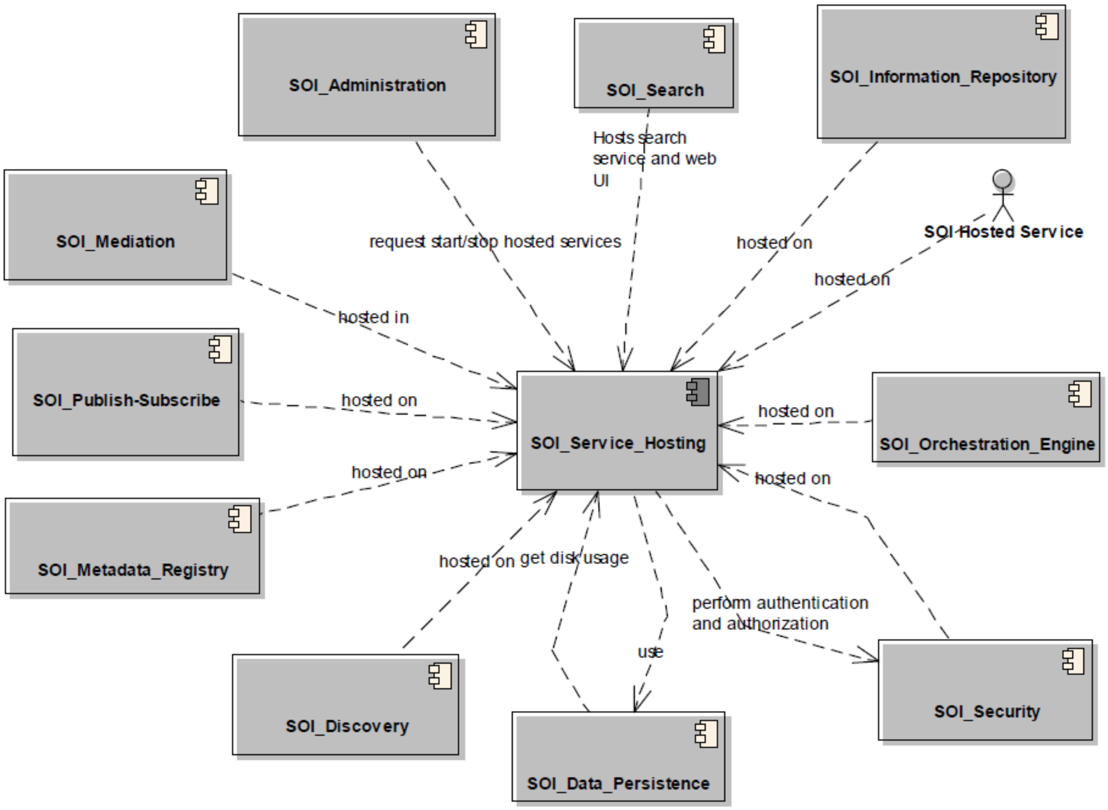


Figure 13. Utility Service Context Diagram

This layer also uses a flexible architecture based on attributes such as loose coupling and asynchronous message passing, emphasizing an incremental approach to adopting and deploying a SOA concept. This design intends to segregate business process into modules that can be easily used again. As an example, the utility service layer takes a schema and an XML document as input, performs the evaluation and reports the result. The same action will be reusable for different schemas that can apply the same policy results.

The basic features that identify the utility service design layer are:

- Data Persistence – provides basic persistence services to SOI components such as; configuration information, search operations, data caching, and data mining.
- Security – provides identification, authentication, authorization and accounting functionality; referred to as “security services.”

- Messaging – provides asynchronous message communication; supports efficient, reliable communication between services within the SOA deployment enclave.
- Discovery - supports service-level integration by providing a service registry to enable clients to publish and locate services.
- Orchestration – provides the capability to compose capabilities from services by validating users and recording start and stop times.
- Notification – provides the ability for software components to send information to other components, users or operators when an event takes place; notifications can be sent via email, chat, instant messaging, etc.
- Publish-Subscribe – provides the capability to publish data and subscribe for data on specified filter criteria. This allows subscribers to filter on a known set of criteria regardless of data type and content.
- Information Repository – processes information object-related task which provides CRUD (Create, Read, Update, Delete) and storage utilization monitoring
- Mediation – provides the capability to perform transformations of information object payloads between the SOI.
- Metadata Registry – enables storing and retrieving information about domain data types and information object types for use in the SOI.
- Search – provides the SOI search functionality such as queries which allows a user/operator to search for persisted information objects.

The utility service layer addresses both non-business centric processing logic and business-specific logic; it results in the redundant implementation of common utility functions across different services. At this layer, utility processing is established which provides reusable utility services for use by other services within the infrastructure inventory. Enterprise components can be exposed as services in this layer, making reuse a real possibility. The utility service layer is dedicated to providing reusable, cross-cutting utility functionality, such as event logging, notification, and exception handling. It is application agnostic in that it can consist of a series of capabilities that draw from multiple enterprise systems and resources, while making this functionality available within a very specific processing context.

2. Entity Service Layer Design

The entity service layer addresses managing specific data types (business entities) using an ESB service for interconnectivity between systems. An analogy would be to look at the same method that a computer's motherboard combines electronic components to create a workstation. At this level, commonalities between service entities exposes services, references other services, and has properties common to all services (Graham, 2004). This layer defines a service-based model for assembling MLS2 key alert objects and defines the 'wiring' that connects the service components. The ESB exposes services, references other services, and has a set of properties. The ESB also defines a way to deploy those assemblies on multiple runtimes within an SOA domain.

The ESB offers the isolation necessary for the evolution of the entity service layer without impacting the consumers. In terms of MLS2s, ESB services assist with communicating between disparate systems and connect the boundary of the task service layer (layer 3) to the entity service layer. Additionally, this layer manages layer 2 business processes by associating ambiguous data types via entity-specific operations. The connection of all the procedures identified between SOA service layers are integrated at this level.

In effort to assist the interoperability of disparate systems, an ESB is built to integrate directly with the SOI. In a TLOC C2 system context, an ESB would typically be used for integrating MLS2s. ESBs support service-level and data-level integration of external systems into the SOI. ESBs are identified as part of system design, in context of the overall system. System analysis identifies the services, data types, and message types that are provided and required by the various external systems, based on tasks associated with the various operator roles. ESB's role is to connect the external system to the SOI to support the required data and message flows and service access. Figure 14 depicts the functionality and interfaces supported by the ESB.

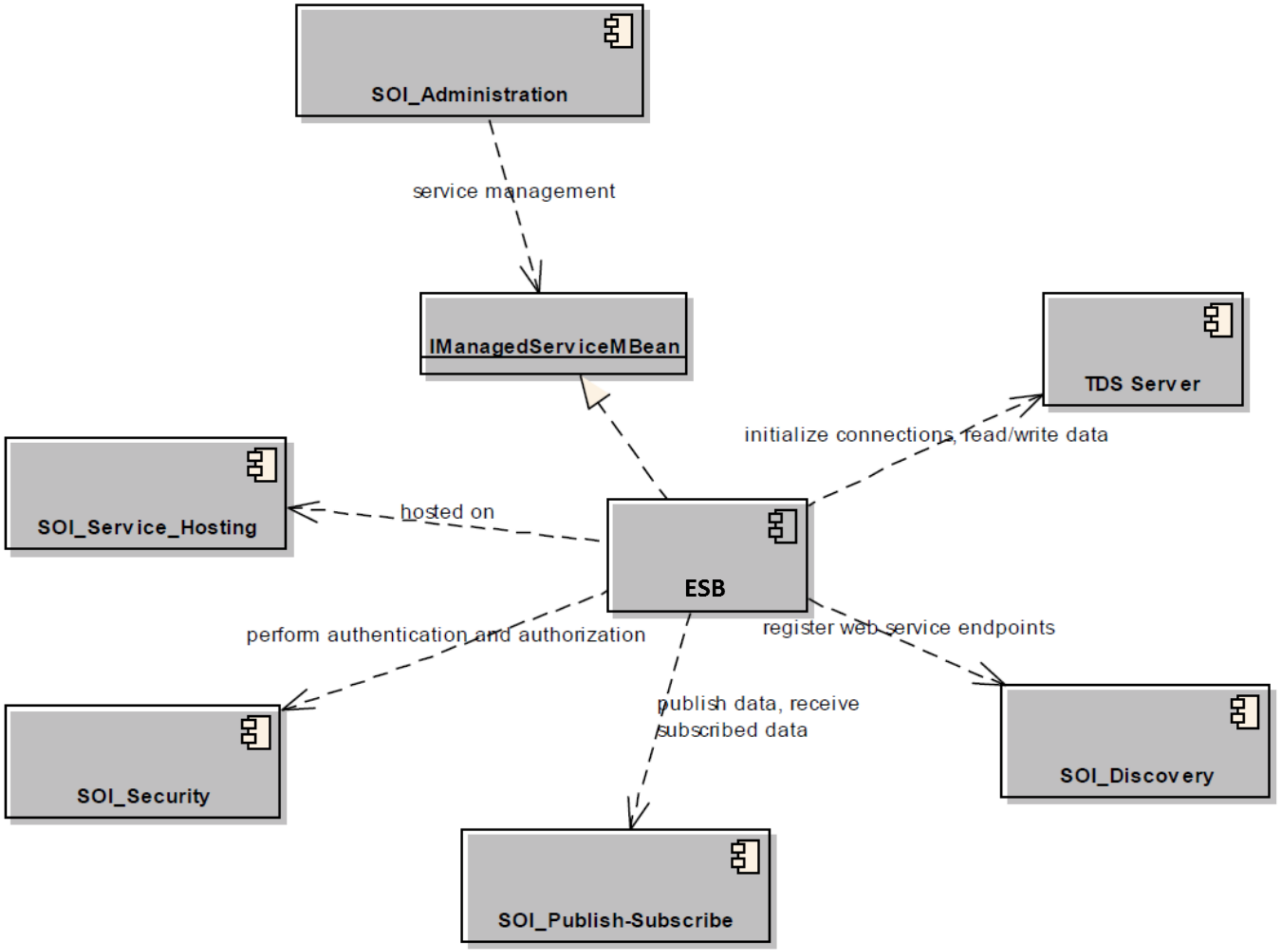

SOI_Administration
service management
IManagedServiceMBean
TDS Server
initialize connections, read/write data
SOI_Service_Hosting
hosted on
ESB
perform authentication and authorization
register web service endpoints
publish data, receive subscribed data
SOI_Security
SOI_Discovery
SOI_Publish-Subscribe

Figure 14. ESB Interface Patterns

ESBs also support workflow-level integration indirectly. The services, messages, and data flows they provide into the SOI can be leveraged via the below listed integration services:

- **Service-level Integration** - The registration of service endpoints provided by the external system with SOI Discovery. In some cases the external system may require specific services and the ESB may assist in locating these service endpoints.
- **Information-level Integration** - The ESB supports data-level integration via a variety of interactions with the SOI Information-level Services:
    - Registering (or confirming registration) data and message formats.
    - Registering data transformations. The intent of the architecture is that ESBs leverage mediation capabilities of the SOI as much as possible rather than performing their own transformations; however, other drivers such as language interoperability or performance may require that some transformations are performed by the ESB.
    - Registering subscriptions for specific data and message formats, receiving data/messages from the SOI based on these subscriptions, and forwarding data to the external system.
- **Management-level Integration** - The ESB supports management-level integration via interactions with the SOI Administration Services:
    - Reports ESB status and the external system status.
    - Reports metrics on service access and the number of data objects sent and received.
    - Management configuration of the ESB - some data interfaces will have configuration options to support enabling/ disabling specific data/message flows or filtering options to control the amount of data flowing in or out of the external system.

3. Task Service Layer Design

Task Service Layer orchestrates other services (entity, task and utilities) and actually performs the business rules (Rob, Kinder, & Graham 2005). Task Services provide complex capabilities oriented at performing a particular task in the domain. For example, in the TLOC C2 domain, Task Services might be full-fledged applications supporting mission planning, intelligence, logistics management, etc. Task Services may leverage the information available at the Entity and Utility Layers while identifying patterns that apply at the individual service level. For the purposes of this architecture, a service could be a hosted MLS2 application; it could also be an application that exposes one or more web service interfaces by registering them in the SOI service registry. Figure 15 depicts various types of services and the interfaces which they expose.

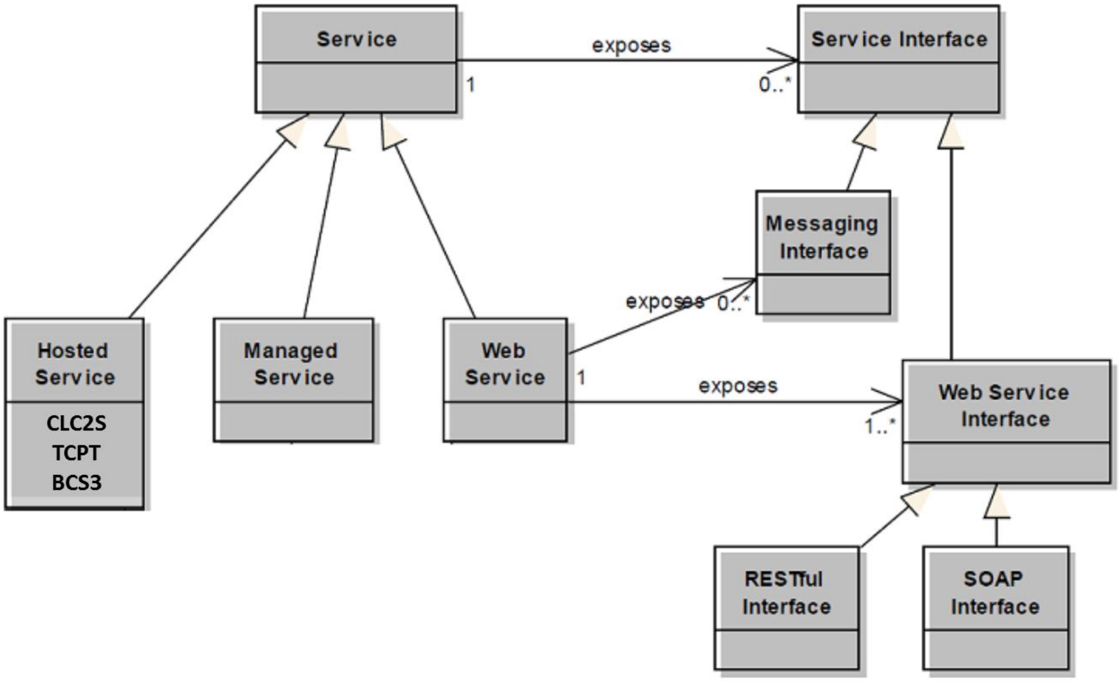


Figure 15. Service Types

As noted in the figure, one of the basic characteristics that identify a software element as a service is the fact that it exposes a service interface. A service may be implemented as a web service-style interface or a messaging-style interface.

- Hosted Service - Given that a service is a separately deployable item, it can either be hosted by the SOI or by the larger system that is integrating the infrastructure. A hosted service is a service whose lifecycle (start/stop) is managed by the SOI. This includes services within the SOI itself such as MLS2 information applications. Most other application-level services are hosted services.
- Managed Service - This is a service which has registered a management interface with the SOI, according to the interface specification defined by SOI Administration.

- Web Service - A web service is a service that exposes a web service interface, such as a Representational State Transfer (style of software architecture for distributed systems such as the World Wide Web). A web service may optionally support a messaging-based interface as well.

The task service layer allows for service abstraction by improving the opportunity to increase the amount of agnostic logic within services based on entity and utility service models. The service abstraction principle is considered valuable because it provides a high level of reuse potential and fully supports the creation of business services by allowing us to cleanly separate and even isolate business process logic into its own domain. This introduces a number of advantages that tie into some of the more strategic benefits of SOA.

C. MLS2 SOA APPROACH OVERVIEW

There are a number of ways that SOA can bring value to an organization. Process optimization has an impact on every aspect of doing business, and savvy organizations are discovering the ways that SOA concepts can bring increased productivity, faster responsiveness, value-added human resources and better corporate compliance. SOA software product line provides the foundation to quickly deliver capabilities to the Marine Corps' operating environment. Through a collaborative organizational structure, SOA leverages various vendors to produce these capabilities. Key engineering artifacts are leveraged to decide which technologies to pursue, document why those technologies were selected, how the technologies affect the users and how the technologies are incorporated into the software product line.

MLS2 SOA software provides the foundation to build service-oriented, mission-relevant products. Through sound software architecture practices, SOA should support the ability to insert new technologies while leveraging existing legacy systems. In addition to the software architecture, the organizational structure and process are crucial to the organization's IT evolution success. MLS2 should employ a SOA approach to provide the ability to link services together and flexibly to add new services in order to meet its evolving operational needs.

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

This thesis presented the qualities of SOA relevant to MLS2 information technologies in order to increase interoperability which is essential in accomplishing C2 for Logistics. The objectives of this research defined the infrastructure and processes required to integrate business entities and software architecture across the MAGTF. The results of this study concluded that the use of a SOA approach can lead to better coordination and interoperability between a disparate IT architecture and accomplishing C2 for Logistics.

The Marine Corps can benefit from the valuable attributes of SOA. The major benefit of implementing a SOA approach is that it allows for reusability of the current software architecture (legacy and current technologies) rather than accepting the status quo and waiting for a long term ERP capability. Additionally, a SOA approach would allow for an architecture that provides a flexible implementation method to meet our current requirements as well as future demands. Implementation of SOA would support and compliment the critical requirements for an ERP solution. SOA would facilitate future ERP requirements and would ensure that performance, availability and interoperability are recognized in future GCSS-MC application increments.

The scope of this thesis was to describe an alternative architecture within SOA layers and principles applicable to MLS2s. Implementing SOA can achieve the following goals to an organization;

- Visibility and flexibility - The emergence of business process management (BPM) promises continuous process improvement and high collaboration between businesses and IT.
- Manage legacy systems - The numerous legacy applications that leave IT departments struggling to reconcile duplicate information, and bits and pieces of business processes strewn across hundreds of applications. SOA addresses these silos and allows organizations to gain better visibility into their data and processes.

- Manage superior data - SOA provides a composite data services platform with a unified set of components for data access, quality, transformation, governance, and caching, among many other data-centric services.
- Reuse services - A related goal of SOA is to effectively manage and reuse enterprise services and data. Services developed by one group in an organization can be used by any other group within or outside the organization if they are published and described in a standards-based format in an accessible registry. When data and services reside with their owners and are shared by consumers as they need it, operational costs associated with their maintenance and management are reduced.
- Align organizational goals - SOA bridge the business and IT gap by enabling continuous process improvement through modeling, simulation, execution, and monitoring in vocabularies that are shared and understood by both business and IT departments.

B. SOA IMPLEMENTATION CHALLENGES

SOA offers a tremendous amount of benefits however, a significant number of SOA projects have failed. Major challenges associated with SOA implementation include three key categories: human, finance, and technical shortfalls. In general, humans do not like change and instinctly are resistant to major modifications. Human interaction with newer technologies usually is a threat to the workforce and creates a sense of losing control or even job security. People get accustomed to and master the use of their older systems and so they are threatened and therefore grow resistant when asked to learn a new system. Another factor that challenges a SOA implementation is funding. Most IT projects often require a large amount of resources. Majority of IT projects fail due to lack of resources which are direct result of overruns in the budget. Form a technology perspective, SOA projects tend to fail due to lack of skilled technology personnel and systems incompatibility. The lack of assigning the right technical experts to a project, leads to implementing incompatible systems. SOA projects require comprehensively skill orientated personnel and consequently involve a significant amount of technical subject matter experts.

C. RECOMMENDATIONS AND FUTURE WORK

This thesis focused on the SOA architectural group of logical layers. These layers assisted with providing analysis and benefits of implementing a SOA approach. BPR system design methods were also used to examine scenario-based data results. There are alternative topics that can be addressed to assist in the overall SOA implementation effort, which include the following;

- Systems Acquisition approach - determine the acquisition processes required to execute SOA; procurement of methodology to be designed within the existing and extensive IT portfolio and the fragmented architecture; from those results predict future IT capabilities needed to meet these requirements.
- ERP approach – determine how SOA could serve to integrate and implement an ERP solution. From the common SOA capabilities, determine required inputs and outputs agnostic of process and then align IT contributions to meet ERP capability requirements.
- Organizational architecture approach - determine the optimal organization and C2 structure (command hierarchy, supported/supporting/adjacent) between TLOCs and the LCEs to execute the mission; what IT capabilities are needed for this optimal organization to execute the mission; explore the impacts of organization structure changes to the complexity of IT capabilities required to execute the mission.

MLS2 SOA solution provides the foundation to build service-oriented, mission-relevant results. Through sound software architecture practices, MLS2 SOA should support the ability to insert new technologies while leveraging existing IT systems. In addition to the software architecture, the organizational structure and business processes are crucial to successful evolution of SOA software solutions.

This thesis presented a coupling model of BPR and SOA in order to satisfy process and technical interoperability aspects of MLS2 agility. The proposed models utilized standards available for mapping BPM concepts via SOA layers, which consisted of three layers: Layer 2 as Service Components; Layer 3 as Services, and; Layer 4 as Business Process. Business improvement approaches, such as BPR is the key to business agility and interoperability. BPR solutions are methods that enable implementation of information systems such as SOA. Current legacy and fragmented IT systems architecture do not satisfy supportability mission objectives. BPR combined with SOA as a design pattern addresses technical agility that satisfies objectives in order to achieve business agility.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Arcitura Education. (2013, July 30). *Elements of service-oriented computing*. Retrieved from Serviceorientation.com: http://serviceorientation.com/serviceorientation

Aziz, S. (2006). *Service oriented architecture: Look beyond the myths to succeed.* (Infosys White Paper). New York, NY: Infosys. Retrieved from http://www.infosys.com/consulting/systems-integration/white-papers/Documents/soa-look-beyond-myths.pdf

Miller, R., & Stanford, F. (2007). *Developing creative computer science graduates for an increasingly competitive market*

Brown, J. et al. (1998). *AntiPatterns: Refactoring software, architectures, and projects in crisis*Hoboken, NJ: Wiley.

Erl, T. (2005). *Service-oriented architecture: Concepts, technology, and design.*Upper Saddle River, NJ: Prentice Hall.

Erl, T. (2007). *SOA: Principles of service design.*Upper Saddle River, NJ: Prentice Hall.

Flurry, G. (2008). *IBM SOA foundation product integration: using websphere transformation extender with ibm enterprise service bus products*. Armonk: NY: IBM developerWorks.

Graham, S. (2004). *Building Web Services with Java: Making Sense of XML, SOAP, WSDL and UDDI* (2nded.).London: Pearson Education.

Hailstone, R. (2009, May). *Technology Audit: Savvion Business Process Management Suite* (Report no. TA001667BPM). Rockville, MD: Butler Group...

Hayes-Roth, R. (2003). *Hyper-beings: how intelligent organizations attain supremacy through information superiority*. Location: Booklocker.com I cannot access this page, but the publisher is Booklocker,co

Josuttis, N. (2007). *SOA in practice: The art of distributed system design.* Sebastopol, CA: O'Reilly Media.

Lessanu, D. (2012, May). SOA Analysis within the Department of Defence architecture framework. *Service Technology Magazine*. Retrieved from http://www.servicetechmag.com/system/application/views/I62/0512-3.pdf

Marks, E. & Bell, M.(2006).Service oriented architecture: A planning and implementation guide forbBusiness andTechnology. Hoboken, NJ: Wiley.

Microsoft, Inc. (2006, December). Learn about software oriented architecture (SOA*)*. Retrieved from http://www.microsoft.com/biztalk/solutions/soa/overview.mspx

Morisio, M. et al. (2002). COTS-based software development: Process and open issues: *The Journal of Systems and Software*, *61*(30), 188-199.

Murch, R. (2004). *Autonomic Computing.*London: Pearson Education.

Potts, M. (2006, January 208). *Seperating Myth from Reality*. Retrieved from SOA world Magazine: http://webservices.sys-con.com/read/175375.htm

Rob, H., Kinder, S., & Graham, S. (2005). *IBM's SOA Foundation: An Architectural Introduction and Overview.* Armpnk, NY: IBM.

Robbins, M. et al. (1998). *Measurement of USMC logistics process: Creating a baseline to support precision logistic implementation*. Santa Monica, CA: RAND.

Ryan K. L. Ko (2009, June). A computer scientist's introductory guide to business process management, *Crossroads*, 15(4)

Soddemann, T. (2004). *Web services and service oriented architectures.*Retrieved from http://www.mpi.nl/delaman/workshop/ppt/soddemann.pdf

The Open Group. (2013). *The Open Group Service-Oriented Architecture (SOA) Work Group: Fostering understanding and adoption of SOA*. Retrieved from The Open Group: http://www3.opengroup.org/getinvolved/workgroups/soa

Thomas, E. (2004). *Service-oriented architecture: a field guide to integrating XML and web services.* Upper Saddle River, NJ: Prentice Hall.

vom Brocke, J.HKVJH & Rosemann, M. (2010), *Handbook on business process management: strategic alignment, governance, people and culture (international handbooks on information systems) (vol. 1).* Berlin: Springer

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California